



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

**ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ**

**Αποτίμηση εγκατάστασης και λειτουργίας  
(κατανεμημένων) συστημάτων διαχείρισης μεγάλου όγκου  
δεδομένων πάνω από συστοιχίες υπολογιστών**

**ΑΧΙΛΛΕΑΣ ΛΥΚΟΤΣΕΤΑΣ**

**Διπλωματική εργασία**

**Επιβλέπων  
Γεώργιος Σταμούλης  
(Καθηγητής Παν. Θεσσαλίας)**

**Λαμία, 2017**



**UNIVERSITY OF THESSALY**  
**SCHOOL OF SCIENCE**  
**INFORMATICS AND COMPUTATIONAL BIOMEDICINE**

**Evaluating the setup and operation of (distributed) big-data  
management systems over large-scale computer clusters**

**ACHILLEAS LYKOTSETAS**

**Master thesis**

**Name of Supervisor**  
**George Stamoulis**  
**(Prof. of University of Thessaly)**

**Lamia , 2017**





**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

**ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ  
ΥΠΟΛΟΓΙΣΤΙΚΗ ΒΙΟΙΑΤΡΙΚΗ**

**«ΠΛΗΡΟΦΟΡΙΚΗ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΣΦΑΛΕΙΑ, ΔΙΑΧΕΙΡΙΣΗ  
ΜΕΓΑΛΟΥ ΟΓΚΟΥ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΠΡΟΣΟΜΟΙΩΣΗ»**

**Αποτίμηση εγκατάστασης και λειτουργίας (κατανεμημένων)  
συστημάτων διαχείρισης μεγάλου όγκου δεδομένων πάνω από  
συστοιχίες υπολογιστών**

**Evaluating the setup and operation of (distributed) big-data management  
systems over large-scale computer clusters**

**ΑΧΙΛΛΕΑΣ ΛΥΚΟΤΣΕΤΑΣ**

**Διπλωματική εργασία**

**Επιβλέπων**

**Γεώργιος Σταμούλης**

**(Καθηγητής Παν. Θεσσαλίας)**

**Λαμία, 2017**

## «Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο **«Αποτίμηση εγκατάστασης και λειτουργίας (κατανεμημένων) συστημάτων διαχείρισης μεγάλου όγκου δεδομένων πάνω από συστοιχίες υπολογιστών»** αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

**Ο Δηλών**

**Αποτίμηση εγκατάστασης και λειτουργίας (κατανεμημένων)  
συστημάτων διαχείρισης μεγάλου όγκου δεδομένων πάνω από  
συστοιχίες υπολογιστών**

**ΑΧΙΛΛΕΑΣ ΛΥΚΟΤΣΕΤΑΣ**

**Τριμελής Επιτροπή:**

Γεώργιος Σταμούλης ( επιβλέπων)  
(Καθηγητής Παν. Θεσσαλίας)

Νέστωρας Ευμορφόπουλος  
(Επίκουρος Καθηγητής Παν. Θεσσαλίας)

Αθανάσιος Λουκόπουλος  
(Επίκουρος Καθηγητής Παν. Θεσσαλίας)

**Επιστημονικός Σύμβουλος:**

Ιωάννης Κωνσταντίνου  
(Λέκτορας Παν. Θεσσαλίας)

## Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιστημονικό υπεύθυνο, **κ. Ιωάννη Κωνσταντίνου**, Λέκτορα του Παν. Θεσσαλίας, για την καθοδήγηση και τη βοήθεια που μου παρείχε για την ολοκλήρωση αυτής της διπλωματικής εργασίας.

Επίσης, θέλω να ευχαριστήσω την οικογένειά μου για την υποστήριξη και την κατανόηση που έδειξε κατά την εκπόνηση αυτής της εργασίας.

Αχιλλέας Λυκοτσέτας

Νοέμβριος 2017

## Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η εγκατάσταση και η βασική διαχείριση ενός Apache Hadoop cluster με τη χρήση του εργαλείου Apache Ambari σε περιβάλλον εικονικοποίησης (virtualization).

Έχοντας ως αφετηρία μία εισαγωγή στη βασική αρχιτεκτονική του Hadoop, προχωράμε στην παρουσίαση της λειτουργίας των επιμέρους τεχνολογιών του (Hadoop Common, Hadoop Distributed File System (HDFS), Hadoop YARN και Hadoop MapReduce).

Ακολουθεί ένας οδηγός εγκατάστασης ενός multi-node cluster , παρουσιάζοντας αναλυτικά τα βήματα που θα ακολουθήσουμε.

Στο τέλος, θα παρουσιαστούν βασικές λειτουργίες διαχείρισης του cluster καθώς και μερικά συμπεράσματα από την χρήση του εργαλείου Apache Ambari.



## **Abstract**

The purpose of this diploma thesis is the installation and basic management of an Apache Hadoop cluster using the Apache Ambari tool in a virtualization environment.

Starting with an introduction to the basic architecture of Hadoop, we are going to present the operation of its individual technologies (Hadoop Common, Hadoop Distributed File System (HDFS), Hadoop YARN and Hadoop MapReduce).

Then follows a wizard for installing a multi-node cluster, detailing the steps we follow.

Finally, basic cluster management functions will be presented, as well as some lessons from using the Apache Ambari tool.

# Πίνακας περιεχομένων

|  |           |
|--|-----------|
| <b>Κεφάλαιο 1 – Εισαγωγή στο Hadoop .....</b>  | <b>1</b>  |
| 1.1 Γενικά .....   | 1         |
| 1.2 Ιστορική αναδρομή .....  | 2         |
| 1.3 Ορισμός Hadoop .....   | 2         |
| 1.4 Σχεδιασμός.....  | 3         |
| 1.5 Λειτουργία.....  | 3         |
| 1.6 Βασικό πλαίσιο.....  | 4         |
| 1.7 Αρχιτεκτονική.....   | 5         |
| 1.7.1 Hadoop Cluster.....  | 6         |
| 1.7.2 HDFS.....  | 8         |
| 1.7.3 MapReduce.....   | 10        |
| 1.7.4 YARN.....  | 12        |
| 1.8 Το οικοσύστημα.....  | 14        |
| 1.9 Άλλα συστήματα αρχείων.....  | 17        |
| 1.10 Άλλες εφαρμογές των στοιχείων του Apache Hadoop .....   | 18        |
| 1.11 Διανομές του Apache Hadoop.....   | 19        |
| 1.12 Εταιρίες που χρησιμοποιούν το Apache Hadoop.....  | 21        |
| <br><b>Κεφάλαιο 2 : Εγκατάσταση Hadoop cluster .....</b>   | <b>23</b> |
| 2.1 Εργαλείο Apache Ambari .....   | 23        |
| 2.2 Πλατφόρμα εικονικοποίησης (virtualization platform) .....  | 23        |
| 2.3 Εγκατάσταση πλατφόρμας εικονικοποίησης και δημιουργία εικονικών μηχανών (virtual machines) ..... | 25        |
| 2.4 Εγκατάσταση Apache Ambari.....   | 26        |

|   |           |
|---|-----------|
| <b>Κεφάλαιο 3: Διαχείριση Hadoop cluster με το εργαλείο Apache Ambari .....</b> | <b>35</b> |
| 3.1 Διαχείριση υπηρεσιών .....  | 35        |
| 3.2 Διαχείριση σταθμών .....  | 38        |
| 3.3 Διαχείριση ελέγχου και ειδοποιήσεων .....                                   | 40        |
| 3.4 Διαχείριση αρχείων και φακέλων .....  | 43        |
| <br><b>Κεφάλαιο 4 : Επίλογος - Συμπεράσματα .....</b>                           | <b>44</b> |
| 4.1 Γιατί Hadoop.....   | 44        |
| 4.2 Γιατί Ambari.....   | 45        |
| 4.3 Συμπεράσματα.....   | 45        |
| <br><b>Πηγές - Βιβλιογραφία .....</b>   | <b>46</b> |

# Κεφάλαιο 1: Εισαγωγή στο Hadoop

## 1.1 Γενικά

Κατά τη διάρκεια των τελευταίων ετών, η βιομηχανία των Big Data έχει γνωρίσει τέτοια τεράστια άνοδο που οι περισσότερες από τις παγκόσμιες εταιρίες επενδύουν όλο και περισσότερο στην έρευνα και την προσέγγισή τους. Με απλά λόγια, ο όρος Big Data περιλαμβάνει τη διαχείριση πολύ μεγάλων όγκων δεδομένων, την επεξεργασία τους με έναν χρονικά αποτελεσματικό τρόπο και την αποθήκευσή τους σε μαζική κλίμακα. Ωστόσο, το αληθινό κίνητρο πίσω από τις μεγάλες επενδύσεις των εταιριών στο συγκεκριμένο τομέα δεν είναι μόνο η συλλογή των δεδομένων αλλά η ανάλυση και η αποσαφήνισή τους.

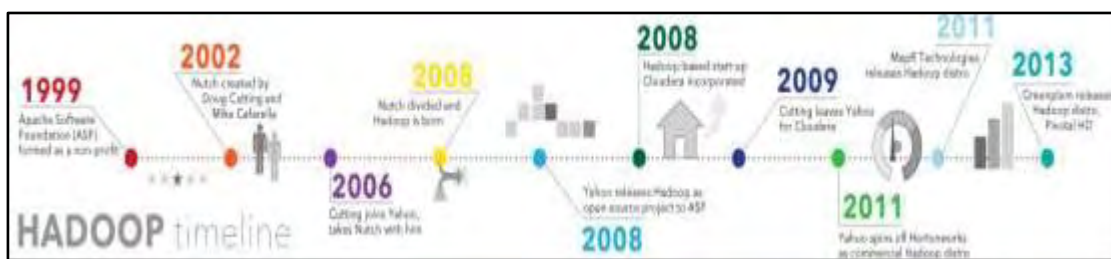
Το Hadoop γνώρισε τεράστια επιτυχία τα τελευταία δύο χρόνια και πλέον είναι αδιαμφισβήτητο το πιο καυτό θέμα των ημερών μας. Τα προηγούμενα χρόνια όλοι μιλούσαν γι' αυτό, ενώ τώρα όλοι καταπιάνονται με αυτό. Μερικοί βρίσκονται στη φάση της εγκατάστασης και της ανάπτυξης και άλλοι στα πρώτα στάδια υλοποίησης και προσομοίωσης. Και φυσικά άλλοι το χρησιμοποιούν ήδη για την αντιμετώπιση πραγματικών και δυσεπίλυτων, μέχρι προηγουμένως, επιχειρηματικών προβλημάτων.

Σήμερα, χρησιμοποιείται ευρέως στις περισσότερες βιομηχανίες, όπως στην οικονομία, στα μέσα ενημέρωσης και ψυχαγωγίας, από κυβερνήσεις, στην υγειονομική περίθαλψη, στις υπηρεσίες πληροφοριών, στο λιανικό εμπόριο και σε άλλες βιομηχανίες με μεγάλες απαιτήσεις δεδομένων. Είναι σχεδιασμένο για την επεξεργασία πολύ μεγάλου όγκου δεδομένων από terabytes σε petabytes και ακόμη μεγαλύτερων.

Το Hadoop είναι η κινητήρια δύναμη πίσω από την ανάπτυξη της βιομηχανίας των Big Data και χωρίς αμφιβολία είναι το “biggest thing” αυτή τη στιγμή αλλάζοντας ριζικά τον τρόπο με τον οποίο οι επιχειρήσεις αποθηκεύουν, επεξεργάζονται και αναλύουν δεδομένα [\[1\]](#).

## 1.2 Ιστορική αναδρομή

Το 2002 οι Doug Cutting και Mike Cafarella έφτιαξαν μία μηχανή αναζήτησης ανοιχτού κώδικα, το Nutch. Η ιδέα εμπνεύστηκε από την ίδια ιδέα που δημιούργησε τη μηχανή αναζήτησης Google και βασιζόταν στην αποθήκευση και επεξεργασία δεδομένων με ένα κατανεμημένο και αυτοματοποιημένο τρόπο έτσι ώστε σχετικά αποτελέσματα αναζήτησης στον ιστό να επιστρέφονται γρηγορότερα. Το 2006 ο Cutting εντάσσεται στη Yahoo, παίρνοντας μαζί του το έργο Nutch και οδηγώντας στην απόσπαση του τμήματος κατανεμημένου υπολογισμού και επεξεργασίας με την ονομασία Hadoop το 2008. Η ονομασία προήλθε από ένα παιχνίδι-ελέφαντα του γιου του Cutting. Ουσιαστικά, το Hadoop ήταν η απόρροια της προσπάθειας από το Yahoo να αντιμετωπίσει το πρόβλημα των υπέρογκων συνόλων δεδομένων, σπάζοντάς τα σε μικρότερα κομμάτια τα οποία θα μπορούσαν να υφίστανται παράλληλη επεξεργασία. Σήμερα, το πλαίσιο και το οικοσύστημα των τεχνολογιών του Hadoop διαχειρίζονται και συντηρούνται από το μη κερδοσκοπικό Apache Software Foundation (ASF) και μία διεθνή κοινότητα προγραμματιστών [2].



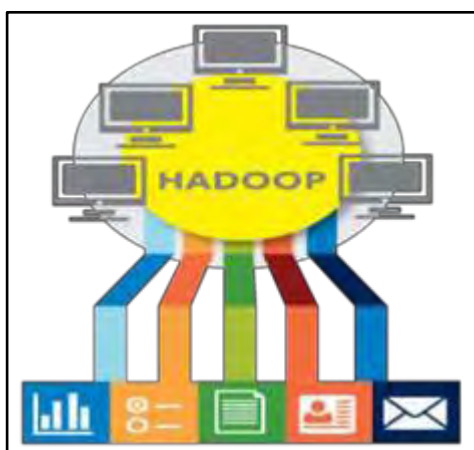
Εικόνα 1 : Ιστορική Αναδρομή

## 1.3 Ορισμός

Το Hadoop είναι ένα προγραμματιστικό πλαίσιο ανοιχτού κώδικα (open-source software framework), γραμμένο σε Java, για την κατανεμημένη αποθήκευση και επεξεργασία πολύ μεγάλων συνόλων, δομημένων και αδόμητων, δεδομένων σε συστάδες υπολογιστών (clusters) με τη χρήση απλών μοντέλων προγραμματισμού. Παρέχει επεκτάσιμη (scalable) και αξιόπιστη μαζική αποθήκευση για κάθε είδος δεδομένων, τεράστια επεξεργαστική ισχύ και την ικανότητα να χειριστεί εικονικά απεριόριστες συντρήουσες εργασίες [3].

## 1.4 Σχεδιασμός

Σε αντίθεση με τα παραδοσιακά συστήματα, το Hadoop επιτρέπει σε πολλαπλές αναλυτικές μονάδες εργασίας (multiple types of analytic workloads) να έχουν πρόσβαση στα ίδια, δομημένα και αδόμητα, δεδομένα, την ίδια χρονική στιγμή. Όλες οι μονάδες του Hadoop έχουν σχεδιαστεί με τη θεμελιώδη παραδοχή ότι οι αποτυχίες υλικού (hardware failures) είναι συχνές και θα πρέπει αυτόματα να αντιμετωπίζονται από το πλαίσιο λογισμικού (framework). Έτσι, αντί να παρέχεται υψηλή διαθεσιμότητα από το υλικό, η βιβλιοθήκη από μόνη της είναι σχεδιασμένη να εντοπίζει και να χειρίζεται τυχόν αποτυχίες σε επίπεδο εφαρμογής, προσφέροντας έτσι υψηλής διαθεσιμότητας υπηρεσίες πάνω από ένα σύμπλεγμα υπολογιστών (cluster), καθένας από τους οποίους είναι επιρρεπής σε αποτυχίες [\[4\]](#).



Εικόνα 2 : Αλληλεπίδραση του Hadoop

## 1.5 Λειτουργία

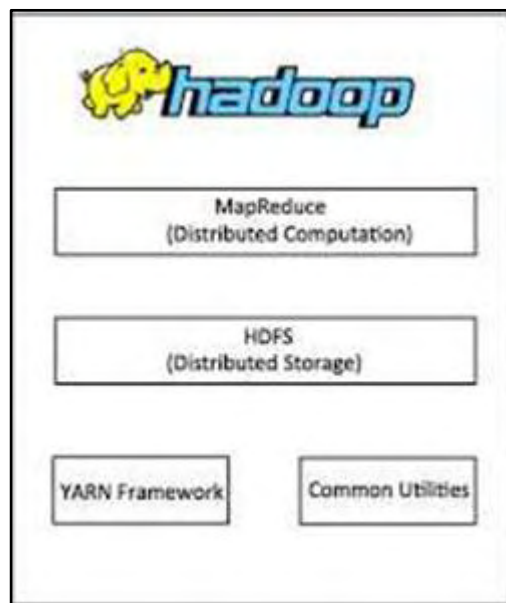
Το Hadoop είναι σχεδιασμένο να λειτουργεί ταυτόχρονα σε ένα μεγάλο αριθμό κόμβων, που δε μοιράζονται μνήμη ή δίσκους.

Έτσι, για την αποθήκευση των δεδομένων, τα αρχεία χωρίζονται σε μεγάλα τεμάχια (blocks) και διανέμονται σε πολλαπλούς κόμβους ενός συμπλέγματος υπολογιστών, με το Hadoop να παρακολουθεί που τοποθετούνται τα δεδομένα. Αν ένας κόμβος βγει εκτός σύνδεσης ή υποστεί βλάβη τα δεδομένα που βρίσκονται αποθηκευμένα σε αυτόν μπορούν να επανακτηθούν από κάποιο άλλο γνωστό, καλό αντίγραφο, αποφεύγοντας, έτσι, πιθανά σφάλματα υλικού [\[5\]](#).

Επιπλέον, για την επεξεργασία των δεδομένων, το Hadoop μεταφέρει στους κόμβους JAR αρχεία (Java Archive files), με βάση τα δεδομένα που χρήζουν επεξεργασίας, έτσι ώστε να τα επεξεργάζονται παράλληλα. Αυτή η προσέγγιση εκμεταλλεύεται την τοπικότητα των δεδομένων (data locality) – οι κόμβοι χειρίζονται μόνο τα δεδομένα στα οποία έχουν πρόσβαση – για να επιτρέψει στο σύνολο των δεδομένων να υποστεί πιο γρήγορη και αποτελεσματική επεξεργασία απ' ό,τι θα μπορούσε σε μία πιο συμβατική αρχιτεκτονική βασιζόμενη σε ένα παράλληλο σύστημα αρχείων (Parallel File System, PFS), όπου υπολογισμοί και δεδομένα διανέμονται μέσω υψηλής ταχύτητας δικτύωσης (high-speed networking).

Αρχιτεκτονικά, ο λόγος που το Hadoop μπορεί να ασχοληθεί με πολλά δεδομένα ταυτόχρονα είναι επειδή τα διαμοιράζει στους διάφορους κόμβους. Και ο λόγος που είναι σε θέση να ζητάει πολύπλοκες υπολογιστικές ερωτήσεις είναι γιατί έχει όλους αυτούς τους επεξεργαστές των κόμβων, να εργάζονται παράλληλα προσδεμένοι μεταξύ τους.

## 1.6 Βασικό πλαίσιο



Εικόνα 3 : Βασικό Πλαίσιο του Hadoop

Ο πυρήνας του Apache Hadoop αποτελείται από τις εξής τέσσερις τεχνολογίες :

- Hadoop Common : Αποτελείται από Java βιβλιοθήκες και βοηθητικά προγράμματα που απαιτούνται για τη λειτουργία των υπολοίπων τμημάτων του Hadoop.
- Hadoop Distributed File System (HDFS) : Ένα κατανεμημένο σύστημα αρχείων σε Java που παρέχει κλιμακώσιμη (scalable) και αξιόπιστη αποθήκευση δεδομένων και υψηλής απόδοσης πρόσβαση σε δεδομένα εφαρμογής.
- Hadoop YARN (Yet Another Resource Negotiator) : Μία πλατφόρμα για τη διαχείριση των πόρων του συμπλέγματος υπολογιστών και το χρονοπρογραμματισμό των εργασιών του χρήστη.
- Hadoop MapReduce : Ένα προγραμματιστικό μοντέλο, βασισμένο στο YARN, για την παράλληλη επεξεργασία μεγάλων συνόλων δεδομένων.

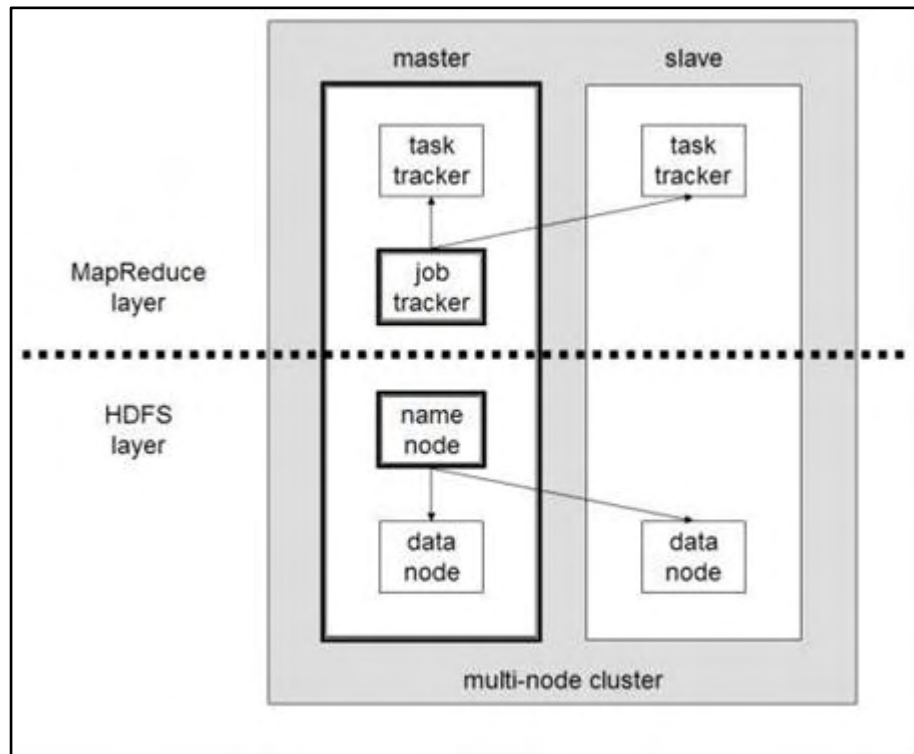
## 1.7 Αρχιτεκτονική

Το βασικό πλαίσιο του Hadoop, όπως αναφέρθηκε προηγουμένως, περιλαμβάνει το Hadoop Common πακέτο, το οποίο παρέχει το σύστημα αρχείων και το στρώμα αφαίρεσης του λειτουργικού συστήματος (OS abstraction layer). Σε αυτό περιέχονται οι απαραίτητες API διεπαφές (ρουτίνες, πρωτόκολλα και εργαλεία) για την ανάπτυξη λογισμικού και εφαρμογών, και τα απαιτούμενα JAR αρχεία και scripts για την εκκίνηση του Hadoop.

Το Hadoop περιλαμβάνει, επίσης, μία MapReduce μηχανή (είτε MapReduce/MR1, είτε YARN/MR2) και ένα σύστημα αρχείων σε επίπεδο χρήστη που διαχειρίζεται τους πόρους αποθήκευσης σε όλο το σύμπλεγμα. Για φορητότητα ανάμεσα σε πλατφόρμες Linux, FreeBSD, Mac OS/X, Solaris και Windows – και ευκολία εγκατάστασης, τα δύο αυτά βασικά συστατικά είναι γραμμένα σε Java και απαιτούν μόνο χαμηλού κόστους υλικό (commodity hardware) [\[6\]](#).

Το Hadoop χρησιμοποιεί master/slave αρχιτεκτονική τόσο για κατανεμημένη αποθήκευση, όσο και για κατανεμημένο υπολογισμό. Ένα μικρό σύμπλεγμα Hadoop περιλαμβάνει ένα κύριο κόμβο (master) και πολλαπλούς κόμβους-εργάτες (slaves). Ο κύριος κόμβος αποτελείται από JobTracker, TaskTracker, NameNode και DataNode. Ένας κόμβος-εργάτης δρα τόσο ως DataNode όσο και ως TaskTracker (Εικόνα 4).





Εικόνα 4: Αρχιτεκτονική Hadoop

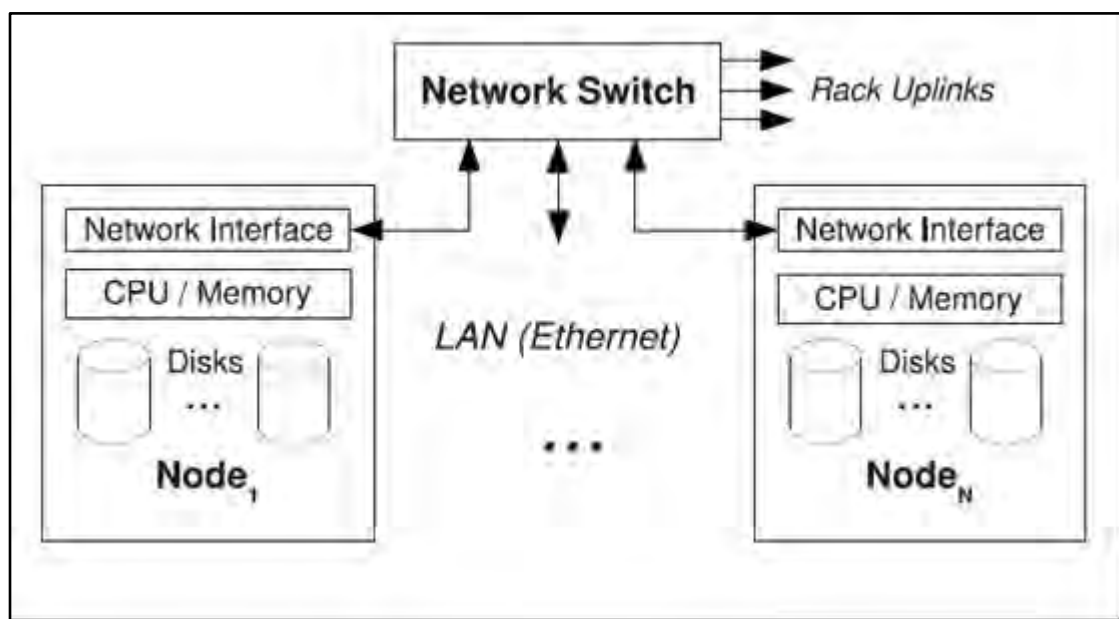
### 1.7.1 Hadoop Cluster

Λόγω της εξαιρετικά παραλληλοποιήσιμης φύσης του υπολογιστικού μοντέλου MapReduce, τα μεγάλα συμπλέγματα υπολογιστών γίνονται ευκολότερα εκμεταλλεύσιμα για την επίτευξη υψηλότερης απόδοσης στην εφαρμογή. Ως εκ τούτου, σε ένα Data-intensive Computing (DC) σύμπλεγμα, η δυναμική κάθε κόμβου μεμονωμένα είναι λιγότερο σημαντική από τη συνολική ικανότητά τους για κλιμάκωση του αριθμού των κόμβων στο σύμπλεγμα. Με βάση αυτή τη φιλοσοφία, τα DC συστήματα, που είναι σχεδιασμένα να τρέχουν υπολογιστικά πλαίσια όπως το Hadoop, έχουν κατασκευαστεί με τις ακόλουθες τεχνολογίες χαμηλού κόστους: επεξεργαστές x86, δίκτυα Ethernet και Serial ATA (SATA) σκληρούς δίσκους. Οποιαδήποτε άλλη επιλογή τεχνολογίας αυξάνει το ανά κόμβο κόστος του συμπλέγματος και τελικά περιορίζει τον αριθμό των κόμβων που μπορούν να αγοραστούν και να χρησιμοποιηθούν, οικονομικά.

Σε ένα τέτοιο σύμπλεγμα, η συνολική απόδοση του συστήματος καθορίζεται από το εύρος ζώνης αποθήκευσης. Κάθε Map ή Reduce εργασία πρέπει να έχει επαρκές διαθέσιμο εύρος ζώνης αποθήκευσης για να ολοκληρώσει αποτελεσματικά το έργο της σε ένα δεδομένο κόμβο υπολογισμού. Η απαίτηση αυτή έχει οδηγήσει σε μια αρχιτεκτονική συμπλέγματος κατά την οποία οι τοπικοί δίσκοι στους κόμβους

υπολογισμού χρησιμοποιούνται ως μέρος ενός κατακευματισμένου συστήματος αρχείων για την αποθήκευση των blocks. Για να εκμεταλλευτεί αποτελεσματικά το εύρος ζώνης των τοπικών δίσκων, το Hadoop επιχειρεί να αναθέσει τις εργασίες στους κόμβους εκείνους όπου είναι αποθηκευμένα τα δεδομένα εισόδου της εκάστοτε εργασίας.

Ένα παράδειγμα αρχιτεκτονικής ενός συμπλέγματος Hadoop σε ένα ενιαίο ράφι (rack) φαίνεται στην Εικόνα 5, όπου κάθε κόμβος υπολογισμού είναι εξοπλισμένος με έναν ή περισσότερους δίσκους και οι κόμβοι διασυνδέονται σε ένα δίκτυο Ethernet. Ένα ράφι είναι πιθανό να είναι διασυνδεδεμένο με έναν ενιαίο, υψηλής απόδοσης διακόπτη Ethernet, που προσφέρει πλήρες εύρος ζώνης μεταξύ όλων των κόμβων εντός του ραφιού. Κάθε ράφι στη συνέχεια συνδέεται με τα υπόλοιπα ράφια μέσω μιας ιεραρχίας διακοπών Ethernet, με πολύ μικρότερο, όμως, εύρος ζώνης μεταξύ των ραφιών, λόγω του κόστους και των περιορισμών καλωδίωσης. Ενώ, κάθε κόμβος μπορεί να επικοινωνεί με οποιονδήποτε άλλο κόμβο (στο ίδιο ή σε διαφορετικό ράφι), υπάρχει πολύ μεγαλύτερο διαθέσιμο εύρος ζώνης μέσα σε ένα ράφι από ό,τι ανάμεσα στα ράφια [\[7\]](#).



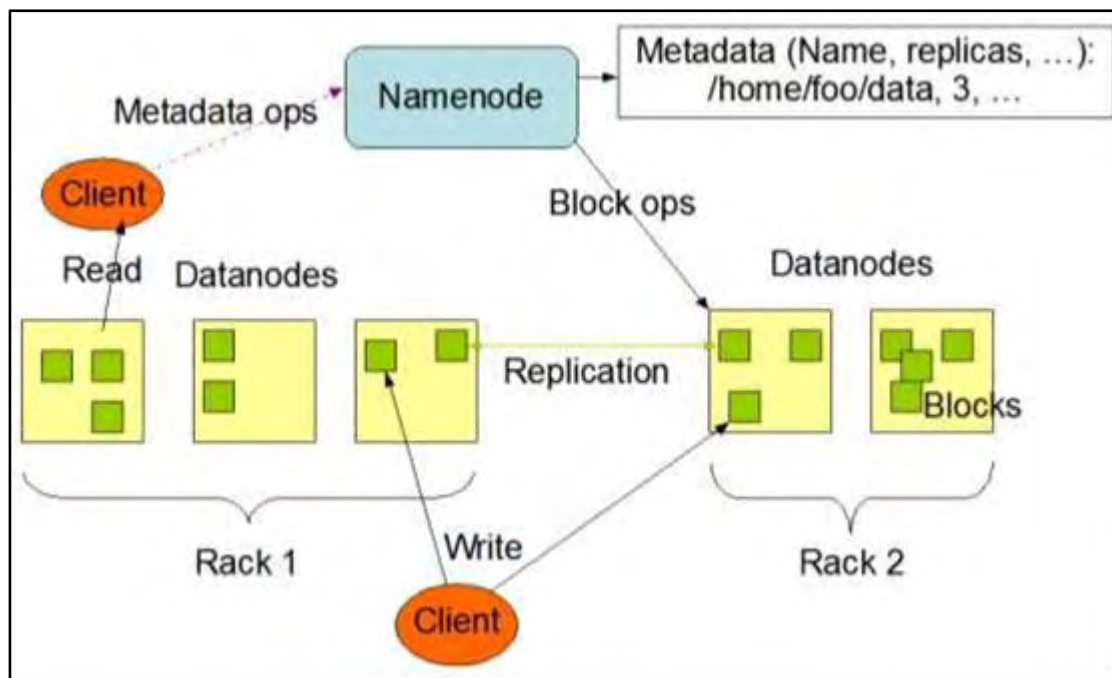
Εικόνα 5 : Αρχιτεκτονική Συμπλέγματος Hadoop

Το Hadoop απαιτεί Java Runtime Environment (JRE) 1.6 ή κάποια νεότερη έκδοση. Τα τυποποιημένα σενάρια (scripts) εκκίνησης και τερματισμού απαιτούν μία SSH σύνδεση (Secure Shell) μεταξύ των κόμβων του συμπλέγματος.

### 1.7.2 HDFS

Το Hadoop Distributed System File είναι ένα κλιμακώσιμο και αξιόπιστο καταναμεμημένο σύστημα αποθήκευσης που συναθροίζει την μνήμη κάθε κόμβου ενός συμπλέγματος υπολογιστών σε ένα ενιαίο σύστημα αρχείων. Το HDFS αποθηκεύει μεμονωμένα αρχεία σε μεγάλα τεμάχια, πετυχαίνοντας έτσι την αποτελεσματική αποθήκευση πολύ μεγάλων και/ή πολυάριθμων αρχείων σε πολλές μηχανές και την παράλληλη πρόσβαση σε μεμονωμένα κομμάτια δεδομένων, χωρίς να χρειάζεται να αναγνωστεί ολόκληρο το αρχείο στη μνήμη ενός και μόνο υπολογιστή. Η αξιοπιστία του επιτυγχάνεται με την αντιγραφή των δεδομένων σε πολλούς και διαφορετικούς κόμβους, με κάθε τεμάχιο δεδομένων να αποθηκεύεται σε τουλάχιστον τρεις. Εάν ένας κόμβος αποτύχει για κάποιο λόγο, τα δεδομένα παραμένουν διαθέσιμα και ένα επιπλέον αντίγραφο των τεμαχίων που κατέχει μπορεί να δημιουργηθεί σε άλλο κόμβο για την προστασία απέναντι σε μελλοντικές αποτυχίες [8].

Το HDFS ακολουθεί την αρχιτεκτονική Master/Slave. Αποτελείται από ένα NameNode (master) και πολλούς DataNodes (slaves). Ο χώρος αποθήκευσης μπορεί να βρίσκεται σε ετερογενή λειτουργικά συστήματα και είναι ενιαίος για όλο το σύμπλεγμα υπολογιστών. Οι πελάτες (clients) μπορούν να βρουν τις τοποθεσίες των τεμαχίων και να προσπελούν τα δεδομένα απευθείας.

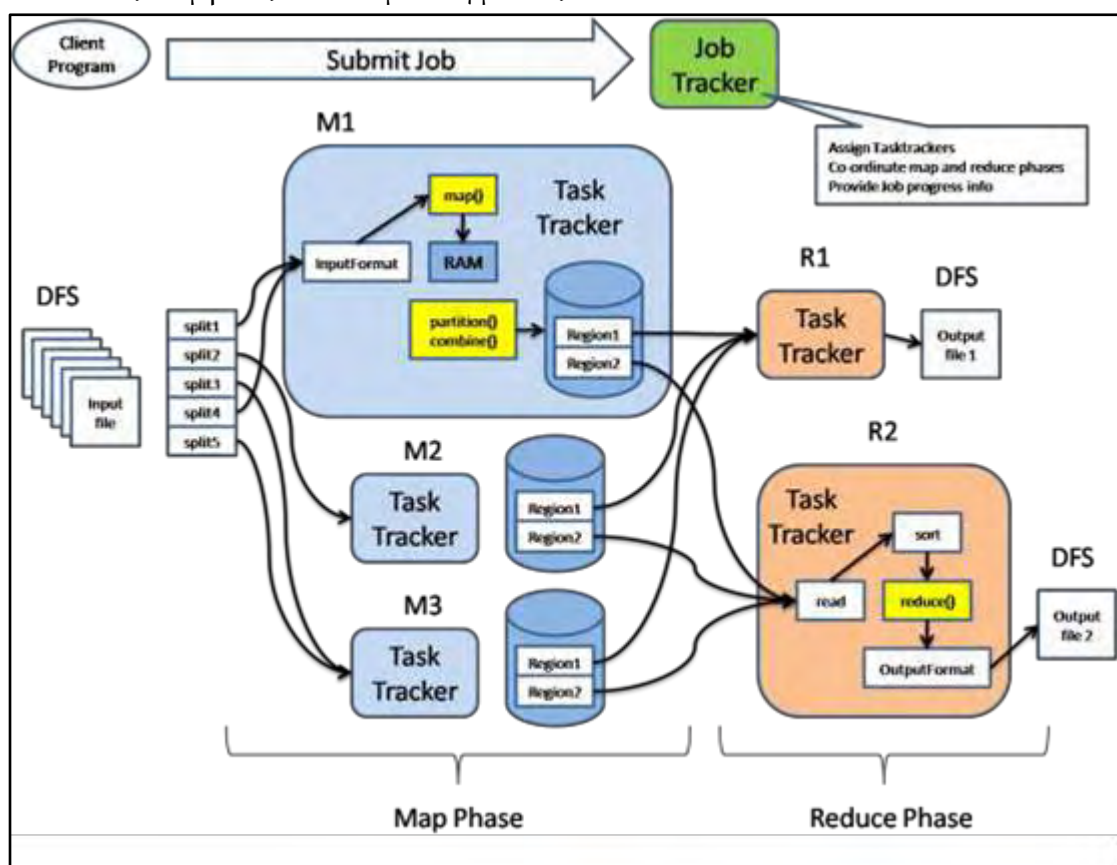


Εικόνα 6 : Αρχιτεκτονική HDFS

Συνολικά, το HDFS εξασφαλίζει διαθεσιμότητα (availability) και υψηλής απόδοσης πρόσβαση (high throughput access) στα δεδομένα. Παρέχει γνωστοποίηση τοποθεσίας (location awareness): το όνομα του ραφιού (του διακόπτη δικτύου ακριβέστερα) όπου βρίσκεται κάθε κόμβος, για τον αποτελεσματικό προγραμματισμό των εργασιών. Φροντίζει για τα αντίγραφα (replication) και την τοπικότητα των δεδομένων (data locality), ενώ παρέχει ανοχή σε σφάλματα (fault tolerance) και διαθέτει την ικανότητα να εξυπηρετεί ένα μεγάλο σύνολο από πελάτες. Χρησιμοποιείται σαν είσοδος δεδομένων και έξοδος αποτελεσμάτων από το MapReduce.

### 1.7.3 MapReduce

Πάνω από το σύστημα αρχείων έρχεται η MapReduce μηχανή, η οποία είναι υπεύθυνη για τη διαχείριση των εργασιών. Όπως φαίνεται στην Εικόνα 7, το σχεδιαστικό μοντέλο γύρω από την υλοποίηση του MapReduce ακολουθεί την αρχιτεκτονική Master/Slave, όπου οι εφαρμογές πελάτη αναθέτουν τις MapReduce εργασίες τους στο JobTracker (Master) και αυτός με τη σειρά του τις μοιράζει στους TaskTrackers (Slaves), που τρέχουν στους κόμβους του συμπλέγματος.



Εικόνα 7 : Αρχιτεκτονική MapReduce

Το Hadoop βελτιστοποιεί αυτόματα την εκτέλεση των MapReduce προγραμμάτων, καθώς σε ένα σύστημα αρχείων με επίγνωση θέσης (location-aware filesystem), όπως το HDFS, ο JobTracker γνωρίζει ποιοι κόμβοι περιέχουν τα δεδομένα, καθώς και ποιοι από αυτούς βρίσκονται σε κοντινή απόσταση. Έτσι, προωθεί την εργασία στους διαθέσιμους TaskTrackers προσπαθώντας να την κρατήσει όσο πιο κοντά γίνεται στα δεδομένα. Εάν ο κόμβος όπου βρίσκονται τα δεδομένα είναι ήδη κατειλημμένος, δίνεται προτεραιότητα στους κοντινότερους κόμβους. Εάν ένας TaskTracker αποτύχει ή εκπνεύσει, η εργασία επαναπρογραμματίζεται. Ο TaskTracker σε κάθε κόμβο δημιουργεί από ένα ξεχωριστό Java Virtual Machine (JVM) για να εκτελέσει την εργασία, με σκοπό να αποφευχθεί η αποτυχία του ίδιου του TaskTracker εάν η τρέχουσα εργασία οδηγήσει σε σφάλμα το JVM. Κάθε TaskTracker επικοινωνεί περιοδικά με το JobTracker μέσω ενός heartbeat μηνύματος για να αναφέρει την πρόοδο διεκπεραίωσης της εργασίας και να ζητήσει πρόσθετες εργασίες όταν βρίσκεται σε περίοδο αδράνειας.

Το MapReduce είναι η καρδιά του Hadoop. Είναι το μοντέλο προγραμματισμού που επιτρέπει στο Hadoop τη μαζική επέκταση σε εκατοντάδες ή χιλιάδες servers σε ένα σύμπλεγμα υπολογιστών και δημιουργήθηκε με σκοπό την εύκολη ανάπτυξη εφαρμογών που επεξεργάζονται γρήγορα, παράλληλα και κατανεμημένα τεράστιες ποσότητες δεδομένων, με έναν αξιόπιστο και ανεκτικό σε βλάβες τρόπο. Το MapReduce σπάει μεγάλα προβλήματα επεξεργασίας δεδομένων σε πολλαπλά στάδια, δηλαδή σε μια σειρά από Map και Reduce βήματα, όπου το καθένα έχει ζευγάρια key/value (κλειδιού/τιμής) για είσοδο και έξοδο και μπορεί να εκτελεστεί την ίδια στιγμή (παράλληλα) σε πολλούς υπολογιστές.

Στο MapReduce μοντέλο, ο υπολογισμός χωρίζεται στα ακόλουθα τμήματα [\[9\]](#):

- Input Reader

Ο αναγνώστης εισόδου χωρίζει την είσοδο σε κατάλληλου μεγέθους «splits» (κομμάτια συνήθως 64-128 MB) και το μοντέλο εκχωρεί ένα split σε κάθε συνάρτηση Map. Ο αναγνώστης εισόδου διαβάζει δεδομένα από κάποιο σταθερό μέσο αποθήκευσης (όπως ένα κατανεμημένο σύστημα αρχείων) και δημιουργεί ζεύγη key/value. Για παράδειγμα, θα διαβάσει ένα φάκελο γεμάτο με αρχεία κειμένου και θα επιστρέψει κάθε πρόταση ως ένα ξεχωριστό αρχείο.

- Map Function

Η συνάρτηση Map λαμβάνει μια σειρά από ζεύγη key/value, επεξεργάζεται το καθένα από αυτά και παράγει μηδενικά ή περισσότερα ζεύγη εξόδου key/value. Οι τύποι εισόδου και εξόδου της Map μπορεί να είναι (και συχνά είναι) διαφορετικοί μεταξύ τους.

Εάν η εφαρμογή κάνει καταμέτρηση λέξεων, η συνάρτηση Map θα σπάσει την πρόταση σε λέξεις και θα εξάγει ένα ζεύγος key/value για κάθε λέξη. Κάθε ζεύγος εξόδου θα περιέχει ως key τη λέξη και ως value τον αριθμό των εμφανίσεων αυτής της λέξης στην πρόταση.

- Partition Function

Κάθε έξοδος της συνάρτησης Map κατανέμεται σε ένα συγκεκριμένο reducer (κόμβος όπου θα εκτελεστεί η συνάρτηση Reduce) από τη συνάρτηση Partition της εφαρμογής, για λόγους θρυμματοποίησης. Η συνάρτηση Partition δέχεται το κλειδί και τον αριθμό των reducers και επιστρέφει το δείκτη (index) του επιθυμητού reducer.

Μεταξύ του Map και του Reduce βήματος, τα δεδομένα ανακατεύονται (shuffle) – ανταλλάσσονται μεταξύ των κόμβων και παράλληλα ταξινομούνται – για να μετακινηθούν από το Map κόμβο, όπου παράχθηκαν, στον κόμβο Reduce.

- Comparison Function

Η είσοδος για κάθε Reduce τραβιέται από το μηχανήμα όπου έτρεξε η Map και ταξινομείται με τη χρήση της συνάρτησης Comparison της εφαρμογής.

- Reduce Function

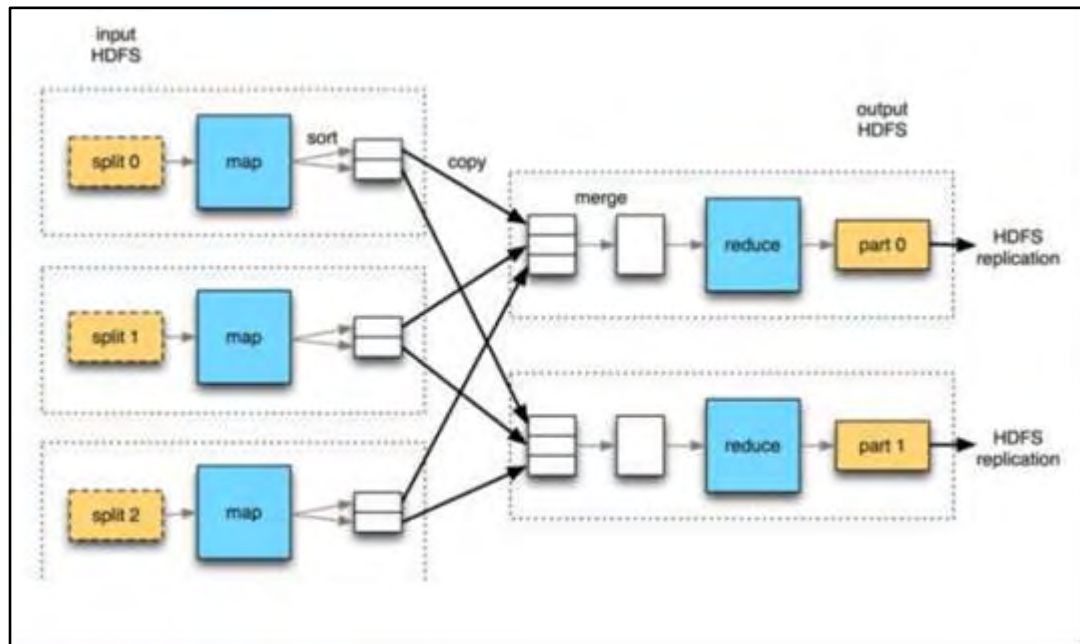
Το μοντέλο καλεί με ταξινομημένη σειρά τη συνάρτηση Reduce της εφαρμογής, μία φορά για κάθε μοναδικό key. Η Reduce μπορεί να επανακληθεί αναδρομικά μέσα από τα values που σχετίζονται με αυτό το key και παράγει μηδέν ή περισσότερες εξόδους.

Στο προηγούμενο παράδειγμα, η συνάρτηση Reduce παίρνει τα values εισόδου, τα συνοψίζει και δημιουργεί μια ενιαία έξοδο της λέξης και του τελικού αθροίσματος.

- Output Writer

Ο συγγραφέας εξόδου εγγράφει την έξοδο από τη Reduce στο HDFS





Εικόνα 8 : Τμήματα MapReduce

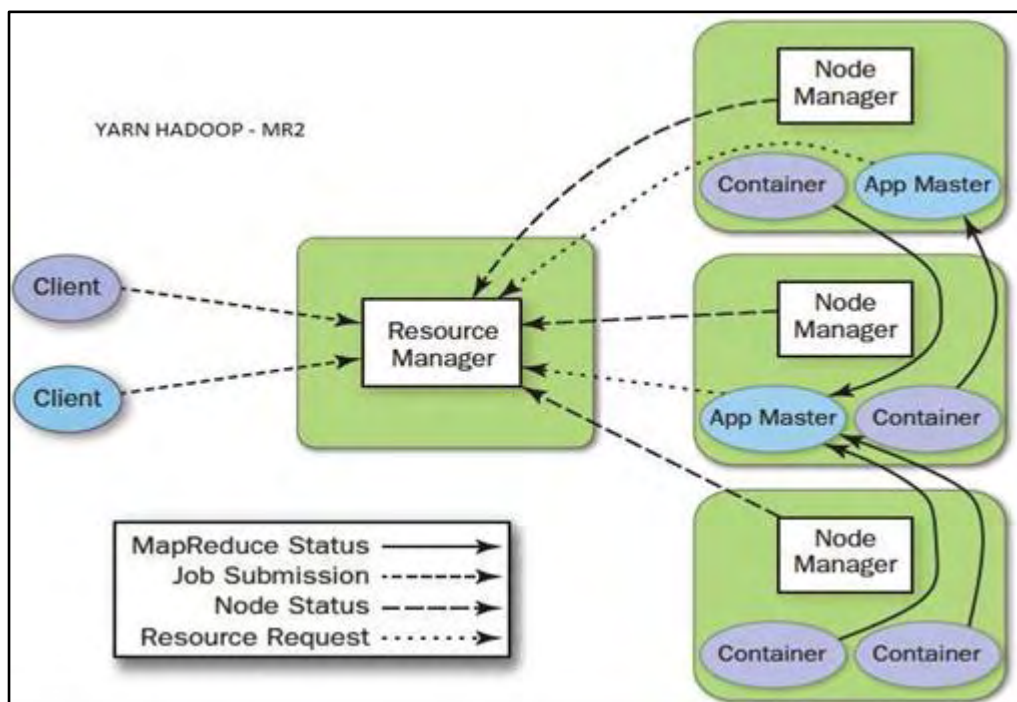
#### 1.7.4 YARN

Στη 2η γενιά του Hadoop, το MapReduce έχει υποστεί μια πλήρη αναμόρφωση και πλέον διαμορφώνεται σε αυτό που αναφέρεται ως MapReduce 2.0 (MRv2) ή YARN.

Η βασική ιδέα πίσω από το YARN είναι να μοιραστούν οι δύο κύριες λειτουργίες του JobTracker, διαχείριση πόρων και χρονοπρογραμματισμός/εποπτεία, σε ξεχωριστές οντότητες, χρησιμοποιώντας έναν ενιαίο ResourceManager και έναν ApplicationMaster ανά εφαρμογή [10].

Ο ResourceManager και ο NodeManager διαμορφώνουν το νέο σύστημα για τη διαχείριση των εφαρμογών με ένα κατανεμημένο τρόπο. Ο ResourceManager είναι η απόλυτη αρχή που διαιτητεύει του πόρους μεταξύ όλων των εφαρμογών του συστήματος. Ο ApplicationMaster είναι μία ειδική οντότητα λογισμικού που διαπραγματεύεται πόρους από το ResourceManager και συνεργάζεται με τους NodeManagers για την εκτέλεση και την παρακολούθηση των συστατικών των εργασιών.

Ο ResourceManager έχει ένα Scheduler, ο οποίος είναι υπεύθυνος για την κατανομή των πόρων στις διάφορες εφαρμογές που τρέχουν στο σύμπλεγμα, σύμφωνα με περιορισμούς όπως χωρητικότητα ουράς και όρια χρηστών. Ο χρονοπρογραμματισμός γίνεται με βάση τις απαιτήσεις κάθε εφαρμογής για πόρους.



Εικόνα 9 : Αρχιτεκτονική YARN

Κάθε ApplicationMaster έχει την ευθύνη για τη διαπραγμάτευση κατάλληλων δοχείων (Container) πόρων από το Scheduler, παρακολουθώντας την κατάστασή τους και την προόδό τους.

Ο NodeManager είναι ο ανά-μηχανή σκλάβος, ο οποίος είναι υπεύθυνος για την δρομολόγηση των Containers των εφαρμογών, παρακολουθώντας τη χρήση των πόρων τους (CPU, μνήμη, δίσκο, δίκτυο) και αναφέροντάς την στο ResourceManager.

Το YARN ενισχύει ένα σύμπλεγμα υπολογισμού με τους ακόλουθους τρόπους:

- Πολυμετοχικότητα (Multi-tenancy)

Επιτρέπει σε πολλαπλές μηχανές πρόσβασης (είτε open-source ή ιδιόκτητες) να χρησιμοποιούν το Hadoop ως το κοινό πρότυπο για batch, διαδραστικές και πραγματικού χρόνου μηχανές που μπορούν να έχουν πρόσβαση ταυτόχρονα στο ίδιο σύνολο δεδομένων.

- Χρησιμοποίηση του συμπλέγματος (Cluster Utilization)

Η δυναμική κατανομή των πόρων του συμπλέγματος βελτιώνει τη χρησιμοποίηση έναντι των περισσότερο στατικών MapReduce κανόνων που χρησιμοποιούνται σε παλαιότερες εκδόσεις του Hadoop.



- Επεκτασιμότητα (Scalability)

Η επεξεργαστική ισχύς των κέντρων δεδομένων συνεχίζει να επεκτείνεται με ταχείς ρυθμούς. Ο ResourceManager του YARN επικεντρώνεται αποκλειστικά στον προγραμματισμό και κρατά ρυθμό όσο το σύμπλεγμα επεκτείνεται σε χιλιάδες κόμβους που διαχειρίζονται petabytes δεδομένων.

- Συμβατότητα (Compatibility)

Το YARN διατηρεί την API συμβατότητα με την προηγούμενη έκδοση του Hadoop. Αυτό σημαίνει ότι υφιστάμενες MapReduce εφαρμογές μπορούν να τρέξουν το YARN χωρίς καμία αναστάτωση στις διεργασίες που ήδη λειτουργούν.

Το YARN είναι η προϋπόθεση για το επιχειρηματικό Hadoop και παρέχει διαχείριση πόρων και μια κεντρική πλατφόρμα όπου προσφέρονται συνεπείς εργασίες, ασφάλεια και εργαλεία διαχείρισης δεδομένων σε συμπλέγματα Hadoop. Επεκτείνει επίσης τη δύναμη του Hadoop σε παλιές και νέες τεχνολογίες που βρίσκονται μέσα στο κέντρο δεδομένων, ώστε να μπορούν να επωφεληθούν από πλευράς κόστους, γραμμικής κλίμακας αποθήκευσης και επεξεργασίας. Είναι το μέλλον του Hadoop.

## 1.8 Το Οικοσύστημα

Ο όρος Hadoop δεν αναφέρεται μόνο στο βασικό πλαίσιο που περιγράφηκε παραπάνω, αλλά και στο οικοσύστημα ή συλλογή επιπρόσθετων πακέτων λογισμικού που μπορούν να εγκατασταθούν παράλληλα με το Hadoop ή στην κορυφή του. Τα εργαλεία αυτά μπορούν να ταξινομηθούν σε Hadoop Data Services (Υπηρεσίες Δεδομένων) και σε Hadoop Operational Services (Λειτουργικές Υπηρεσίες) με βάση την λειτουργικότητα που προσφέρουν [\[11\]](#).

Hadoop Data Services είναι τα εργαλεία εκείνα που επιτρέπουν στους χρήστες να χειρίζονται και να επεξεργάζονται δεδομένα ευκολότερα. Μερικά από αυτά είναι:

- Apache Hive

Το Hive είναι υποδομή αποθήκης δεδομένων, χτισμένη στην κορυφή του Hadoop για την παροχή σύνοψης δεδομένων, ad-hoc query και ανάλυσης μεγάλων συνόλων δεδομένων. Παρέχει ένα μηχανισμό για μετάδοση δομής πάνω στα δεδομένα και για την αναζήτηση αυτών χρησιμοποιώντας μία γλώσσα τύπου SQL που ονομάζεται HiveQL

(HQL). Το Hive διευκολύνει την επικοινωνία μεταξύ του Hadoop και διαφόρων εργαλείων απεικόνισης και business intelligence (BI).

- Apache Pig

Το Pig επιτρέπει τη συγγραφή σύνθετων MapReduce μεταμορφώσεων, χρησιμοποιώντας μια απλή scripting γλώσσα που ονομάζεται Pig Latin. Η Pig Latin καθορίζει ένα σύνολο μεταμορφώσεων σε ένα σύνολο δεδομένων, όπως άθροισμα, join και ταξινόμηση. Το Pig μεταφράζει την Pig Latin σε Hadoop MapReduce έτσι ώστε να μπορεί να εκτελεστεί μέσα στο HDFS.

- Apache HCatalog

Το HCatalog είναι ένα στρώμα διαχείρισης πινάκων και αποθηκών για το Hadoop, που επιτρέπει σε χρήστες με διαφορετικά εργαλεία επεξεργασίας δεδομένων – συμπεριλαμβανομένων των Hive, Pig και MapReduce – να πραγματοποιούν εγγραφές και αναγνώσεις δεδομένων ευκολότερα. Οι αφαιρέσεις πινάκων του HCatalog παρουσιάζουν στους χρήστες μια σχεσιακή όψη των δεδομένων στο HDFS και διασφαλίζουν ότι οι χρήστες δεν χρειάζεται να ανησυχούν για το που ή σε ποια μορφή αποθηκεύονται τα δεδομένα τους.

- Apache HBase

Το HBase είναι μία μη σχεσιακή βάση δεδομένων που τρέχει στην κορυφή του HDFS. Παρέχει ανεκτική σε σφάλματα αποθήκευση και γρήγορη πρόσβαση σε μεγάλες ποσότητες αραιών δεδομένων. Προσθέτει, επίσης, στο Hadoop συναλλακτικές δυνατότητες, επιτρέποντας στους χρήστες να διεξάγουν ενημερώσεις, εισαγωγές και διαγραφές.

- Apache Sqoop

Το Sqoop είναι ένα εργαλείο σχεδιασμένο για την αποτελεσματική μεταφορά αδόμητων δεδομένων μεταξύ Hadoop και δομημένων αποθηκών, όπως σχεσιακές βάσεις δεδομένων. Εισάγει δεδομένα από εξωτερικές πηγές, είτε απευθείας στο HDFS είτε σε συστήματα όπως το Hive και το HBase. Το Sqoop μπορεί επίσης να τραβήξει δεδομένα από το Hadoop και να τα εξάγει σε εξωτερικά μέσα αποθήκευσης, όπως σχεσιακές βάσεις δεδομένων και αποθήκες δεδομένων.

- Apache Flume

Το Flume είναι μια υπηρεσία για την αποτελεσματική συλλογή, συνάθροιση και μετακίνηση μεγάλων ποσοτήτων δεδομένων καταγραφής (log data). Επιτρέπει σε δεδομένα καταγραφής από πολλές διαφορετικές πηγές, όπως web servers, να αποθηκεύονται εύκολα σε ένα κεντρικό HDFS αποθετήριο.

- Apache Solr

Το Solr είναι ένα επεκτάσιμο εργαλείο αναζήτησης που προσφέρει ευρετηρίαση, αξιοπιστία, κεντρική διαμόρφωση, failover και ανάκτηση δεδομένων αποθηκευμένων στο HDFS. Το Solr ενεργοποιεί τις δυνατότητες αναζήτησης και πλοήγησης στις περισσότερες από τις μεγαλύτερες ιστοσελίδες του διαδικτύου, επιτρέποντας ισχυρή αναζήτηση πλήρους κειμένου και σχεδόν πραγματικού χρόνου ευρετηρίαση

- Apache Zeppelin

Το Zeppelin είναι ένα open-source, web-based εργαλείο που επιτρέπει analytics και απεικόνιση μεγάλων συνόλων δεδομένων σε κατανεμημένα, διαδραστικά και συνεργατικά συστήματα, όπως το Hadoop. Παρέχει απορρόφηση, εξερεύνηση, οπτικοποίηση και κοινή χρήση των δεδομένων.

Hadoop Operational Services είναι τα εργαλεία εκείνα τα οποία έχουν προκύψει για να βοηθήσουν στις λειτουργίες και τη διαχείριση ενός συμπλέγματος Hadoop, όπως:

- Apache Ambari

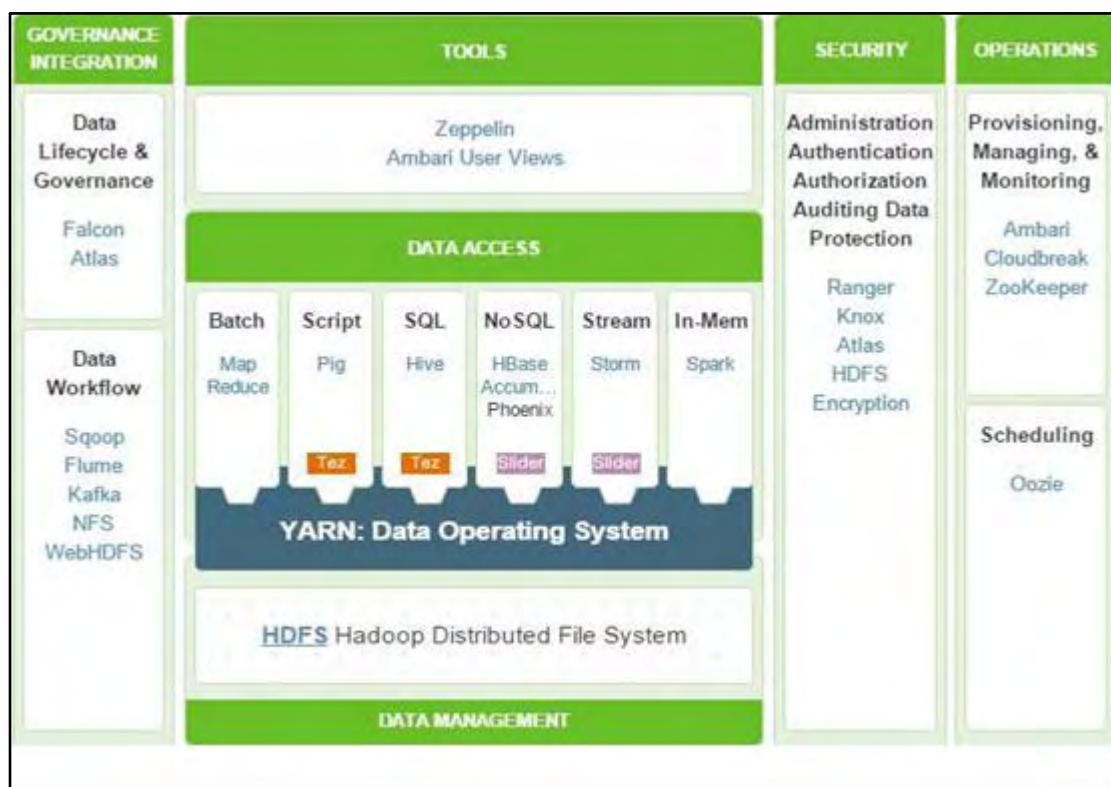
Το Ambari παρέχει ένα διαισθητικό σύνολο εργαλείων για την παρακολούθηση, τη διαχείριση και τον αποτελεσματικό εφοδιασμό ενός συμπλέγματος Hadoop. Απλοποιεί τη λειτουργία και κρύβει την πολυπλοκότητα του Hadoop, κάνοντάς το να δουλεύει ως μια ενιαία, συνεκτική πλατφόρμα δεδομένων.

- Apache Oozie

Το Oozie είναι μία web εφαρμογή σε Java, που χρησιμοποιείται για το χρονοπρογραμματισμό των εργασιών του Hadoop. Συνδυάζει πολλαπλές εργασίες διαδοχικά σε μία λογική μονάδα εργασίας και υποστηρίζει Hadoop εργασίες για το MapReduce, Pig, Hive και Sqoop.

- Apache ZooKeeper

Το ZooKeeper παρέχει λειτουργικές υπηρεσίες για ένα σύμπλεγμα Hadoop, συμπεριλαμβανομένης μίας κατανεμημένης υπηρεσίας διαμόρφωσης, μίας υπηρεσίας συγχρονισμού και ενός μητρώου ονομάτων. Αυτές οι υπηρεσίες επιτρέπουν στις κατανεμημένες διαδικασίες του Hadoop να συντονιστούν πιο εύκολα μεταξύ τους.



Εικόνα 10 : Οικοσύστημα Hadoop

## 1.9 Άλλα συστήματα αρχείων

Το Apache Hadoop υποστηρίζει επίσημα αρκετά άλλα συστήματα αρχείων τα οποία είναι τα εξής:

- Το Amazon S3 σύστημα αρχείων.

Αυτό είναι στοχευόμενο και σχεδιασμένο για τα cluster που αναπτύσσονται στην server on-demand υποδομή του Amazon Elastic Compute Cloud. Στο συγκεκριμένο σύστημα αρχείων επειδή τα αρχεία βρίσκονται απομακρυσμένα, το Hadoop χάνει τα πλεονεκτήματα που προσφέρουν στην απόδοση του, η τοπικότητα των δεδομένων και η επίγνωση της τοποθεσίας των δεδομένων. Εδώ δεν υπάρχει επίγνωση της τοποθεσίας

(rack) των δεδομένων και υπάρχει καθυστέρηση μέχρι να είναι σίγουρο ότι μεταφέρθηκαν τα δεδομένα χωρίς σφάλμα.

- To CloudStore (Kosmos Distributed File System),

το οποίο προσφέρει επίγνωση της τοποθεσίας (rack) και ήταν μια C++ υλοποίηση του Google File System. Το CloudStore προσφέρει επεκτασιμότητα, αναπαραγωγή και αντιγραφή των δεδομένων σε πολλαπλούς κόμβους για ασφάλεια, έλεγχο για την ακεραιότητα των δεδομένων, fail-over από την πλευρά του client και πρόσβαση από Java, Python αλλά και C++. Τέλος, προσφέρει και την δυνατότητα φόρτωσης (mount) του σε λειτουργικό σύστημα Linux μέσω μίας μονάδας (module) FUSE.

- To FTP σύστημα αρχείων.

Σε αυτή την περίπτωση τα αρχεία απλά βρίσκονται σε απομακρυσμένους FTP server και είναι προσβάσιμα μέσω αυτών.

- To HDFS (το οποίο αναλύθηκε εκτενώς παραπάνω).

Το HDFS είναι το προκαθορισμένο κατανεμημένο σύστημα αρχείων του Hadoop και έχει όπως αναφέραμε ήδη επίγνωση της τοποθεσίας των δεδομένων και σχεδιάστηκε να επεκτείνεται σε δεκάδες petabyte δεδομένων και εκτελείται πάνω από το σύστημα αρχείων του λειτουργικού συστήματος που χρησιμοποιείται.

- To maprfs σύστημα αρχείων της

Το maprfs σχεδιάστηκε από την αρχή για να παρέχει υψηλή διαθεσιμότητα, σωστά στιγμιότυπα και αντίγραφα, παρέχοντας παράλληλα υψηλότερη επεκτασιμότητα από το HDFS και υψηλότερη απόδοση. Το maprfs είναι μέρος της διανομής της MapR και ως το προκαθορισμένο σύστημα αρχείων στο Elastic Map Reduce της Amazon και στην Google Compute Engine.

- Τα μόνο για ανάγνωση HTTP και HTTPS συστήματα αρχείων.

## 1.10 Άλλες εφαρμογές των στοιχείων του Apache Hadoop

Το HDFS σύστημα αρχείων δεν χρησιμοποιείται αποκλειστικά για εκτέλεση εργασιών MapReduce. Έχει και άλλες εφαρμογές, πολλές από τις οποίες είναι και αυτές project της Apache και ανήκουν στο γενικότερο οικοσύστημα του Hadoop και λειτουργούν πάνω από αυτό προσφέροντας επιπλέον δυνατότητες, στα οποία θα αναφερθούμε παρακάτω. Μερικά από αυτά που συνεργάζονται με το HDFS είναι, η βάση δεδομένων HBase, το Apache Mahout σύστημα μηχανικής μάθησης και το Apache Hive σύστημα αποθήκης δεδομένων. Επίσης, το Hadoop μπορεί να χρησιμοποιηθεί για οποιαδήποτε εργασία που έχει υψηλές απαιτήσεις σε δεδομένα, χρειάζεται πολύ επεξεργασία (αλλά όχι σε πραγματικό χρόνο) και που μπορεί να ολοκληρωθεί χωρίζοντας και επεξεργάζοντας τα κομμάτια των δεδομένων παράλληλα. Πολλές τέτοιας φύσης εφαρμογές στις οποίες έχει χρησιμοποιηθεί το Hadoop είναι:

- Ανάλυση αρχείων καταγραφής.
- Ανάλυση για λόγους marketing, για την βοήθεια στην λήψη αποφάσεων, για έρευνα των τάσεων της αγοράς σε κάποιο τομέα κλπ.
- Προχωρημένη εξόρυξη δεδομένων και μηχανική μάθηση.
- Επεξεργασία εικόνων.
- Επεξεργασία μηνυμάτων XML.
- Επεξεργασία ελεύθερου κειμένου (π.χ. Twitter) και περιήγηση του web (web crawling)
- Γενική αρχειοθέτηση ακόμη και σχεσιακών ή σε μορφή πινάκων δεδομένων για λόγους συμβατότητας.

## 1.11 Διανομές του Apache Hadoop

Υπάρχει ένα μεγάλος αριθμός εταιριών που προσφέρουν εμπορικές εφαρμογές και διανομές και/ή παρέχουν υποστήριξη για το Hadoop. Κάποιες από αυτές ακολουθούν παρακάτω:

- CDH (Cloudera's Distribution Including Apache Hadoop) της Cloudera.[\[12\]](#)

Η CDH είναι η πιο ολοκληρωμένη, δοκιμασμένη και ευρέως διαδεδομένη διανομή του Apache Hadoop. Η CDH είναι 100% ανοιχτού κώδικα και είναι η μόνη λύση Hadoop που

προσφέρει μαζική επεξεργασία, διαδραστική SQL διαδραστική αναζήτηση, καθώς και επιχειρησιακού (enterprise) επιπέδου συνεχή διαθεσιμότητα. Πιο πολλές επιχειρήσεις έχουν κατεβάσει την CDH από ότι όλες τις άλλες διανομές μαζί. Αυτό το ισχυρό εργαλείο αυτοματοποίησης προσφέρει τον πιο γρήγορο και τον πιο εύκολο τρόπο για να στήσεις το cluster σου και να εξερευνήσεις τις πρώτες σου περιπτώσεις χρήσης. Η CDH προσφέρει τα βασικά στοιχεία το Hadoop – επεκτάσιμη αποθήκευση και κατανεμημένη επεξεργασία – αλλά και τις απαραίτητες επιχειρησιακές δυνατότητες, όπως ασφάλεια, υψηλή διαθεσιμότητα και ενσωμάτωση με ένα ευρύ φάσμα λύσεων υλικού και λογισμικού. Όλη αυτή η εργασία ενσωμάτωσης γίνεται αυτόματα για τον χρήστη και όλη η λύση της Cloudera είναι διεξοδικά δοκιμασμένη και πλήρως τεκμηριωμένη, βοηθώντας στην λύση πραγματικών επιχειρησιακών προβλημάτων.

Η CDH διανομή της Cloudera, αποτελεί μέρος διαφορετικών εκδόσεων της λύσης της Cloudera, οι οποίες είναι: η Cloudera Standard, η Cloudera Enterprise, το Cloudera Manager το οποίο είναι μέρος και των δύο απλά χωρίς όλες τις δυνατότητες διαθέσιμες στην Cloudera Standard και ο Cloudera Navigator που είναι ένα αυτοματοποιημένο σύστημα διαχείρισης των δεδομένων.

- Η MapR διανομή του Apache Hadoop της MapR Technologies Inc.[13]

Η διανομή MapR της εταιρίας ισχυρίζεται ότι προσφέρει πλήρη προστασία των δεδομένων, κανένα μοναδικό σημείο αστοχίας, βελτιωμένη απόδοση και δραματικά πλεονεκτήματα στην ευκολία της χρήσης. Η MapR συνεισφέρει στο Apache Hadoop και στα project που σχετίζονται με αυτό όπως το Apache HBase, το Apache Pig, το Apache Hive και το Apache Zookeeper. Η MapR, επίσης έκανε μία συμφωνία με την EMC Corporation τον Μάιο του 2011 και υποστηρίζει μία διανομή του Apache Hadoop που είναι συγκεκριμένη για τα συστήματα της EMC. Επιπλέον, επιλέχτηκε από την Amazon για να προσφέρει μια αναβαθμισμένη και πιο αξιόπιστη έκδοση της υπηρεσίας Amazon Elastic Map Reduce (EMR) που είναι μέρος των υπηρεσιών Amazon Web Services (AWS) και προσφέρει τις M3 και M5 εκδόσεις της MapR ως premium επιλογές. Γενικά, έχει επιλεγεί από πολλές εταιρείες λόγω της αποδεδειγμένης υψηλής απόδοσης της σε συγκριτικές δοκιμές με άλλες κατανεμημένες/παράλληλες υλοποιήσεις επεξεργασίας δεδομένων αλλά και με το ίδιο το Apache Hadoop και με άλλες διανομές αυτού. Η MapR προσφέρει τρεις εκδόσεις της διανομής τους γνωστές ως M3, M5 και M7. Η M3 είναι μία ελεύθερη έκδοση της M5 αλλά με μειωμένες δυνατότητες και λειτουργίες αλλά πάλι πολύ ισχυρές και ικανές για το ξεκίνημα και την ενασχόληση με το MapReduce και φυσικά με την ταχύτητα της MapR. Η M7 είναι σαν την M5 αλλά προσθέτει μία ειδικά



κατασκευασμένη υλοποίηση του HBase, που υλοποιεί το HBase API απευθείας στο επίπεδο του συστήματος αρχείων.

- Η HDP (Hortonworks Data Platform) της Hortonworks Inc. [14]

Η διανομή της Hortonworks, Apache Hadoop Data Platform είναι 100% ανοιχτού κώδικα, πλήρως δοκιμασμένη, πιστοποιημένη και έχει σχεδιαστεί με στόχο την ενσωμάτωση με τις υπάρχουσες data center τεχνολογίες και προσφέρει την επαναχρησιμοποίηση των υπάρχοντων πόρων και ικανοτήτων. Γι' αυτό και προσφέρει συμβατότητα με πολλά συστήματα. Επίσης, η εταιρία είναι από τους μεγαλύτερους συνεισφέροντες στο Apache Hadoop και σε όλα τα project που σχετίζονται. Η έκδοση 2 της HDP (η πιο πρόσφατη) προσφέρει επεξεργασία και αποθήκευση με μεγάλη επεκτασιμότητα και πολλαπλές εργασίες (workloads), εφαρμογές και επεξεργαστικές μηχανές με μεγαλύτερη απόδοση με την βοήθεια της δεύτερης έκδοσης του MapReduce (MR2) το YARN (η μόνη διανομή προς το παρόν που συμπεριλαμβάνει όλες τις βελτιώσεις του YARN), του HDFS2 και της υψηλής διαθεσιμότητας και ασφάλειας που παρέχει. Επίσης, προσφέρει την διανομή της για Windows και είναι η μόνη που προσφέρει κάτι τέτοιο. Ενσωματώνεται εύκολα με την επιχειρησιακή λογική και το οικοσύστημα της Microsoft και είναι διαθέσιμη για Windows Server είτε στο Windows Azure Cloud. Επιπλέον, η λύση της Hortonworks ενσωματώνει σχεδόν όλα τα συναφή project του Hadoop, για να προσφέρει μια πλατφόρμα με όλα τα βασικά στοιχεία απαραίτητα για τις λειτουργικές υπηρεσίες των δεδομένων που απαιτούνται από μία επιχείρηση και προσφέρει πραγματική πολλαπλή χρήση των δεδομένων, χάρη στην δυνατότητα που έχει να αλληλεπιδρά ταυτόχρονα με πολλούς τρόπους με τα δεδομένα.

- Το HDInsight της Microsoft που προσφέρεται στο Windows Azure.[15]

Το HDInsight είναι μία υπηρεσία του Windows Azure που προσφέρει το Apache Hadoop στο Cloud, παρέχοντας την δυνατότητα προβολής και διορατικότητας πάνω στα δεδομένα μέσω του Excel, εύκολη διαχείριση και μία διανομή του Hadoop έτοιμη για λειτουργία για κάθε επιχείρηση. Επίσης, προσφέρει πλούσιες δυνατότητες για την ανάπτυξη εφαρμογών για την χρήση της μέσω του γνωστού .NET framework, της JAVA και άλλες γλώσσες προγραμματισμού.



## 1.12 Εταιρίες που χρησιμοποιούν το Apache Hadoop

- [Yahoo! \[16\]](#)

Η Yahoo! Inc. , το Φεβρουάριο του 2008, παρουσίασε μία από τις μεγαλύτερες εφαρμογές του Hadoop μέχρι τότε, το Yahoo! Search Webmap, το οποίο είναι μία εφαρμογή που «τρέχει» σε ένα cluster με πάνω από δέκα χιλιάδες επεξεργαστικούς πυρήνες πάνω σε λειτουργικό σύστημα Linux και παράγει δεδομένα που βοηθούν και χρησιμοποιούνται σε κάθε ερώτημα αναζήτησης προς την Yahoo!. Υπάρχουν πολλά Hadoop cluster στην Yahoo! και κανένα HDFS σύστημα αρχείων ή MapReduce εργασία, χωρίζεται σε πολλαπλά datacenter. Κάθε κόμβος ενός Hadoop cluster, όταν ξεκινά, φορτώνει την εικόνα του Linux συμπεριλαμβανομένου και της διανομής του Hadoop. Μία από τις εργασίες που εκτελούν τα cluster τις Yahoo! είναι ο υπολογισμός των δεικτών για την μηχανή αναζήτησης της. Τον Ιούνιο του 2009, η Yahoo! έκανε διαθέσιμο τον πηγαίο κώδικα, της έκδοσης του Hadoop που λειτουργεί στα πραγματικά της cluster, και συνεχίζει να συνεισφέρει στο Hadoop και την κοινότητα του με ότι αλλαγή κάνει. Επιπλέον, οι προγραμματιστές της εταιρίας διορθώνουν σφάλματα (bugs) που έχει το project, παρέχουν βελτιώσεις στην σταθερότητα του Hadoop και παρέχουν αυτό τον κώδικα ώστε και άλλοι χρήστες να επωφεληθούν από την εργασία που κάνει αρχικά για δικό της όφελος.

- [Facebook \[17\]](#)

Το 2010, το Facebook ισχυριζόταν ότι έχει το μεγαλύτερο Hadoop cluster στον κόσμο με είκοσι ένα Petabyte δεδομένων και συνεχίζει να αυξάνεται σε μέγεθος μέχρι και σήμερα. Τον Ιούνιο του 2012 τα δεδομένα είχαν φτάσει τα εκατό Petabyte και τον Νοέμβριο του 2012 ανακοίνωσαν ότι η αποθήκη των δεδομένων τους αυξάνεται κάθε μέρα κατά μισό Petabyte περίπου.

- [Άλλοι χρήστες/εταιρίες \[18\]](#)

Το Hadoop χρησιμοποιείται, πλέον από πάρα πολλές εταιρείες οι μισές των οποίων είναι στην λίστα Fortune 50 (λίστα με τις πιο κερδοφόρες και ισχυρές εταιρίες). Βέβαια σε κάθε εταιρία χρησιμοποιείται για διαφορετικό σκοπό και σε διαφορετική έκταση με cluster μερικών δεκάδων κόμβων, μέχρι cluster με χιλιάδες κόμβους και με Petabyte δεδομένων προς επεξεργασία. Μερικές από αυτές είναι η A9.com (Amazon), η Adobe, η AOL, η Cloudspace, το Cornell University Web Lab, το eBay, το Greece.com , η Hulu, η IBM, η ImageShack, το Last.fm, το LinkedIn, το Mail.gr, η NAVTEQ Media Solutions, η The New York Times, η Rackspace, το Spotify, το StumbleUpon, η Telefonica Research, το Twitter, το University of Glasgow, το University of Maryland, το Veoh και πολλές άλλες.

## Κεφάλαιο 2 : Εγκατάσταση Hadoop cluster

### 2.1 Εργαλείο Apache Ambari

Το Apache Ambari project έχει ως στόχο να κάνει την διαχείριση του Hadoop πολύ πιο απλή αναπτύσσοντας λογισμικό για την ρύθμιση, και παραμετροποίηση του Hadoop, για την διαχείριση του Hadoop και για την επίβλεψη και παρακολούθηση των cluster του Hadoop. Το Ambari παρέχει, μία διαισθητική και εύκολη στην χρήση web διεπαφή, υποστηριζόμενη από τα RESTful API του και έχει συμβατότητα με τα: HDFS, MapReduce, Hive, HCatalog, HBase, Zookeeper, Oozie, Pig, και Sqoop. Το Ambari, βοηθά τους διαχειριστές του συστήματος βήμα προς βήμα να εγκαταστήσουν και να ρυθμίσουν νέες υπηρεσίες στο cluster σε οποιοδήποτε αριθμό κόμβων.

Παρέχει κεντρική διαχείριση για ξεκίνημα, τερματισμό και παραμετροποίηση των υπηρεσιών για όλο το cluster. Έχει ένα κεντρικό πίνακα για παρακολούθηση της υγείας και της κατάστασης όλου του cluster. Χρησιμοποιεί το Ganglia σύστημα παρακολούθησης για συλλογή μετρικών του συστήματος και το Nagios για προειδοποιήσεις για κρίσιμες καταστάσεις για το σύστημα (αστοχία κόμβου, χαμηλό υπόλοιπο του δίσκου κ.α.). Επίσης, βοηθά τους προγραμματιστές εφαρμογών να χρησιμοποιήσουν τα REST API του Ambari για να ενσωματώσουν όλα τα πλεονεκτήματα του στην εφαρμογή τους.

### 2.2 Πλατφόρμα εικονικοποίησης (virtualization platform)

Για την δημιουργία των εικονικών μηχανών επιλέχθηκε το Proxmox Virtual Environment 5.0 , το οποίο είναι ένα περιβάλλον εικονικοποίησης ανοιχτού κώδικα που χρησιμοποιεί τον πυρήνα του Linux και είναι βασισμένο στην διανομή Debian, υπό την έκδοση GNU AGPL, v3.

Το Proxmox VE χρησιμοποιεί τη θύρα (port) 8006 για την πρόσβαση στο περιβάλλον διαχείρισης, ενώ παράλληλα διαθέτει REST API για την χρήση του από άλλες εφαρμογές. Επίσης, οι μορφές εικονικοποίησης που υποστηρίζει είναι τα containers (LXC), καθώς και την πλήρη εικονικοποίηση μέσω του KVM. Όσον αφορά την εγκατάστασή του, υπάρχει η δυνατότητα χειροκίνητης παραμετροποίησης του συστήματος, εγκαθιστώντας τα κατάλληλα πακέτα και δημιουργώντας τα κατάλληλα διαμερίσματα στον δίσκο, αλλά και απευθείας εγκατάσταση του μέσω αρχείου εικόνας .iso στο δίσκο.

Ένα από τα βασικότερα πλεονεκτήματα χρήσης του είναι το εύχρηστο περιβάλλον του. Η απλή και κατανοητή του σχεδίαση προσφέρει στον χρήστη την ευκολία να δημιουργήσει εικονικές μηχανές και containers σε μικρό χρόνο, ενώ παράλληλα μέσω της διαχειριστικής πλατφόρμας έχει την δυνατότητα να συνδεθεί στο τερματικό του κάθε εικονικού μηχανήματος, να διαχειριστεί τις εικονικές μηχανές ή τα containers, το storage ή το cluster που έχει δημιουργήσει.

Το Proxmox VE αποτελεί μία Open Source ολοκληρωμένη λύση για εικονικοποίηση και βασίζεται στην εικονικοποίηση KVM και στην container-based εικονικοποίηση, και διαχειρίζεται KVM virtual machines, Linux containers (LXC), αποθηκευτικούς χώρους, εικονικοποιημένα δίκτυα και HA Clusters.

Τα σημαντικότερα χαρακτηριστικά συνοψίζονται στα εξής :

- Open Source Server Virtualization with KVM and LXC

Ο ανοιχτού κώδικα Hypervisor KVM αποτελεί μία ολοκληρωμένη λύση Linux σε συστήματα X86, δίνοντας τη δυνατότητα να τρέχουν πολλαπλές εικονικές μηχανές με linux ή Windows images.

- Container-based virtualization

Τα Containers αποτελούν μία εναλλακτική λύση εικονικοποίησης προσφέροντας χαμηλότερη επιβάρυνση.

- Linux Containers (LXC)

Το linux containers είναι ένα περιβάλλον επιπέδου εικονικοποίησης λειτουργικών συστημάτων που επιτρέπει να τρέχουν πολλαπλά linux συστήματα σε ένα μοναδικό linux host. Οι χρήστες του συστήματος linux μπορούν να δημιουργήσουν και να διαχειριστούν containers συστημάτων ή εφαρμογών μέσω εργαλείων και API.

- Live Migration

Δυνατότητα μετακίνησης εικονικών μηχανών από ένα φυσικό μηχάνημα σε ένα άλλο χωρίς να υπάρχει διακοπή λειτουργίας.

- Central Management

Μέσω του ενσωματωμένου web-based interface υπάρχει η δυνατότητα διαχείρισης των virtual machines, των containers, ακόμη και ολόκληρου του cluster που έχει δημιουργηθεί.

Επίσης το Proxmox VE χρησιμοποιεί το Proxmox Cluster File System (pmxcfs), ένα database file system για την αποθήκευση των αρχείων παραμετροποίησης, αποθηκεύοντας έτσι για εκατοντάδες εικονικές μηχανές τα configuration files. Χρησιμοποιώντας το corosync αυτά τα αρχεία μπορούν να αναπαραχθούν σε πραγματικό χρόνο σε όλους τους κόμβους του Cluster.

Το ενσωματωμένο εργαλείο backup (vzdump) δημιουργεί συνεχή snapshots των KVM virtual machines και των containers. Είναι επίσης συμβατό με συστήματα αποθήκευσης όπως το NFS, iSCSI LUN, Ceph RBD ή το Sheepdog.

- Proxmox VE High Availability Cluster

Με το Proxmox VE HA Cluster ενεργοποιείται η έννοια της υψηλής διαθεσιμότητας. Έτσι μέσω της πλατφόρμας διαχείρισης Proxmox VE HA Manager παρακολουθούνται όλα τα virtual machines και τα containers του cluster και ενεργοποιείται αυτόματα σε περίπτωση αποτυχίας ενός τμήματος του cluster.

Το μοντέλο αποθήκευσης της πλατφόρμας Proxmox VE είναι αρκετά ευέλικτο υποστηρίζοντας δικτυακής αποθήκευσης επιπέδου ( π.χ. LVM Group (network backing with iSCSI targets), iSCSI target, NFS Share, Ceph RBD, Direct to iSCSI LUN, GlusterFS), καθώς και τοπικής αποθήκευσης (π.χ. LVM Group (local backing devices like block devices, FC devices, DRBD, etc.), Directory (storage on existing filesystem), ZFS. [\[19\]\[20\]](#)

## 2.3 Εγκατάσταση πλατφόρμας εικονικοποίησης και δημιουργία εικονικών μηχανών (virtual machines)

Για την υλοποίηση του cluster χρησιμοποιήσαμε τρεις (3) blade servers της σειράς Hewlett Packard ProLiant BL G1 που είναι εγκατεστημένοι σε ένα HP c7000 chassis με τα παρακάτω χαρακτηριστικά:

| Μοντέλο            | CPUs   | Μνήμη | Σκληρός δίσκος    |
|--------------------|--|-------|-------------------|
| ProLiant BL685c G1 | 4x Dual-Core AMD Opteron, 2800 MHz (8 total threads) | 16 GB | 146 GB SCSI RAID1 |
| ProLiant BL465c G1 | 2x Dual-Core AMD Opteron, 2800 MHz (4 total threads) | 6 GB  | 72 GB SCSI RAID1  |
| ProLiant BL465c G1 | 2x Dual-Core AMD Opteron, 2800 MHz (4 total threads) | 6 GB  | 72 GB SCSI RAID1  |

Πίνακας 1

Σε κάθε φυσικό server εγκαταστήσαμε ξεχωριστά την πλατφόρμα εικονικοποίησης ProxMox VE [21] και στη συνέχεια δημιουργήσαμε τις τρεις (3) ιδεατές μηχανές (virtual machines) όπου εγκαταστήσαμε το λειτουργικό σύστημα Linux της διανομής GNU Debian 7.1 [22].

Η επιλογή αυτής της διαδικασίας εγκατάστασης εκμεταλλεύεται τη δυνατότητα λήψης στιγμιότυπων (snapshots) των ιδεατών μηχανών έτσι ώστε η εγκατάσταση να γίνει βήμα-βήμα και να υπάρχει η δυνατότητα επαναφοράς σε προηγούμενη κατάσταση σε περίπτωση κάποιου λάθους παραμετροποίησης.

## 2.4 Εγκατάσταση Apache Ambari

Αρχικά, για την εγκατάσταση του εργαλείου Apache Ambari ρυθμίζουμε κατάλληλα τους servers που θα αποτελέσουν το cluster:

### 1ο βήμα:

Ενημερώνουμε τις εκδόσεις των πακέτων του λειτουργικού του συστήματος:

```
apt-get update -y && apt-get upgrade -y
```

### 2ο βήμα:

Ρυθμίζουμε τα αρχεία */etc/hosts* με τις αντιστοιχίες των IP διευθύνσεων και των hostnames των server του cluster. Το αρχείο θα πρέπει να έχει τις ίδιες εγγραφές σε όλους τους servers και στη περίπτωσή μας θα έχει τη μορφή:

```
195.130.79.200 hc-ambari.teiste.gr hc-ambari
195.130.79.201 hc-node01.teiste.gr hc-node01
195.130.79.202 hc-node02.teiste.gr hc-node02
```

### 3ο βήμα:

Εγκαθιστούμε το πακέτο εξυπηρετητή συγχρονισμού NTP (Network Time Protocol) server:

```
apt-get install ntp -y
```

Προσθέτουμε τον εξυπηρετητή συγχρονισμού *ntp.teiste.gr*.

### 4ο βήμα:

Το Hadoop για να λειτουργήσει θα πρέπει οι υπολογιστές που αποτελούν το cluster να μπορούν να συνδέονται μεταξύ τους με την χρήση ssh χωρίς κωδικό. Αυτό επιτυγχάνεται με την χρήση ζεύγους ιδιωτικού/δημόσιου κλειδιού. Η βασική ιδέα της

τεχνικής αυτής είναι ότι ο κάθε χρήστης μπορεί να έχει ένα ιδιωτικό κλειδί (αρχείο) το οποίο έρχεται ζεύγος με ένα δημόσιο κλειδί. Ο χρήστης τοποθετεί το δημόσιο κλειδί στους υπολογιστές τους οποίους θέλει να έχει πρόσβαση, και με την χρήση του ιδιωτικού μπορεί να συνδέεται σε αυτούς χωρίς κωδικό. Η πιστοποίηση του χρήστη γίνεται καθώς μόνο ο χρήστης έχει στην κατοχή του το ιδιωτικό κλειδί.

Δημιουργούμε στον server όπου θα εγκαταστήσουμε το Apache Ambari ένα ιδιωτικό κλειδί `id_rsa` και ένα δημόσιο κλειδί `id_rsa.pub`:

```
ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
```

και στη συνέχεια εισάγουμε το δημόσιο κλειδί `id_rsa.pub` στον κατάλογο με τα αποδεκτά δημόσια κλειδιά του χρήστη `root` του server:

```
cat /root/.ssh/id_rsa.pub > /root/.ssh/authorized_keys
```

Τέλος μεταφέρουμε τον κατάλογο `.ssh` στον χρήστη `root` και των υπόλοιπων servers του cluster:

```
scp -r .ssh/ root@hc-node01.teiste.gr:~
```

```
scp -r .ssh/ root@hc-node02.teiste.gr:~
```

#### 5<sup>ο</sup> βήμα:

Κατεβάζουμε κι ενημερώνουμε τα αποθετήρια της έκδοσης του Apache Ambari:

```
wget -O /etc/apt/sources.list.d/ambari.list
```

```
http://public-repo-1.hortonworks.com/ambari/debian7/2.x/updates/2.5.1.0/ambari.list
```

```
apt-get update -y
```

#### 6<sup>ο</sup> βήμα:

Εγκαθιστούμε το πακέτο του Apache Ambari:

```
apt-get install ambari-server
```

Η διαδικασία είναι αυτοματοποιημένη και ολοκληρώνετε σε λίγο χρονικό διάστημα.

#### 7<sup>ο</sup> βήμα:

Ρυθμίζουμε τον server:

```
ambari-server setup
```

Σε όλα τα ερωτήματα επιλέγουμε Yes.

Η επαλήθευση της σωστής εγκατάστασης γίνεται με την εντολή:

```
ambari-server status
```

που στην περίπτωση μας είναι:

```
root@hc-ambari:~# ambari-server status
```

```
Using python /usr/bin/python
```

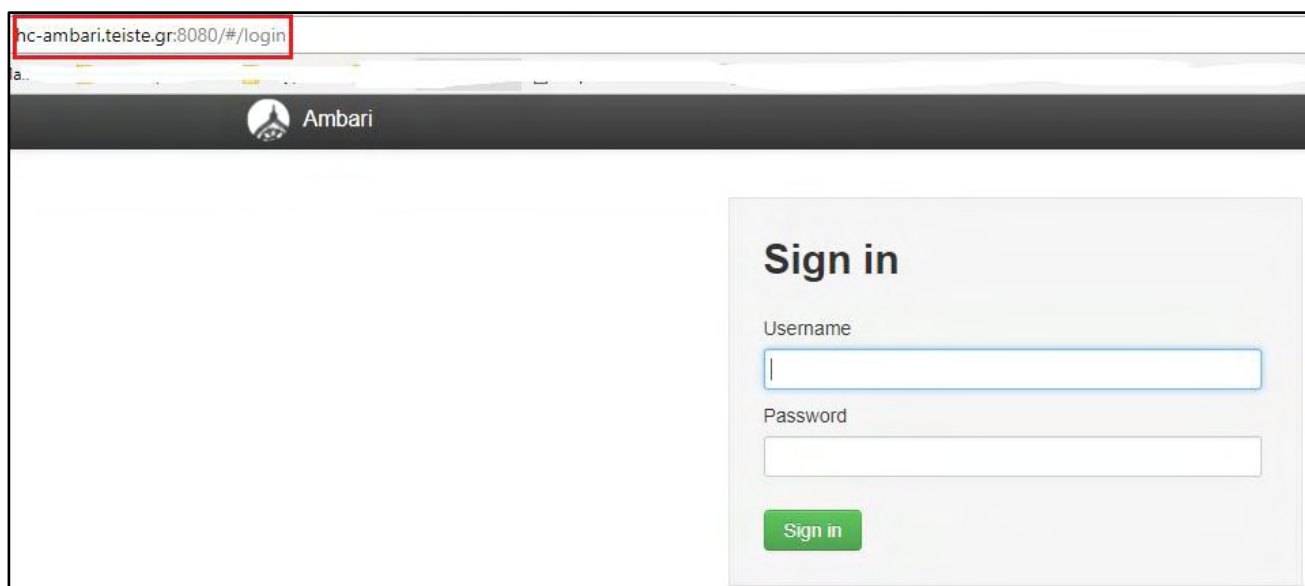
```
Ambari-server status
```

```
Ambari Server running
```

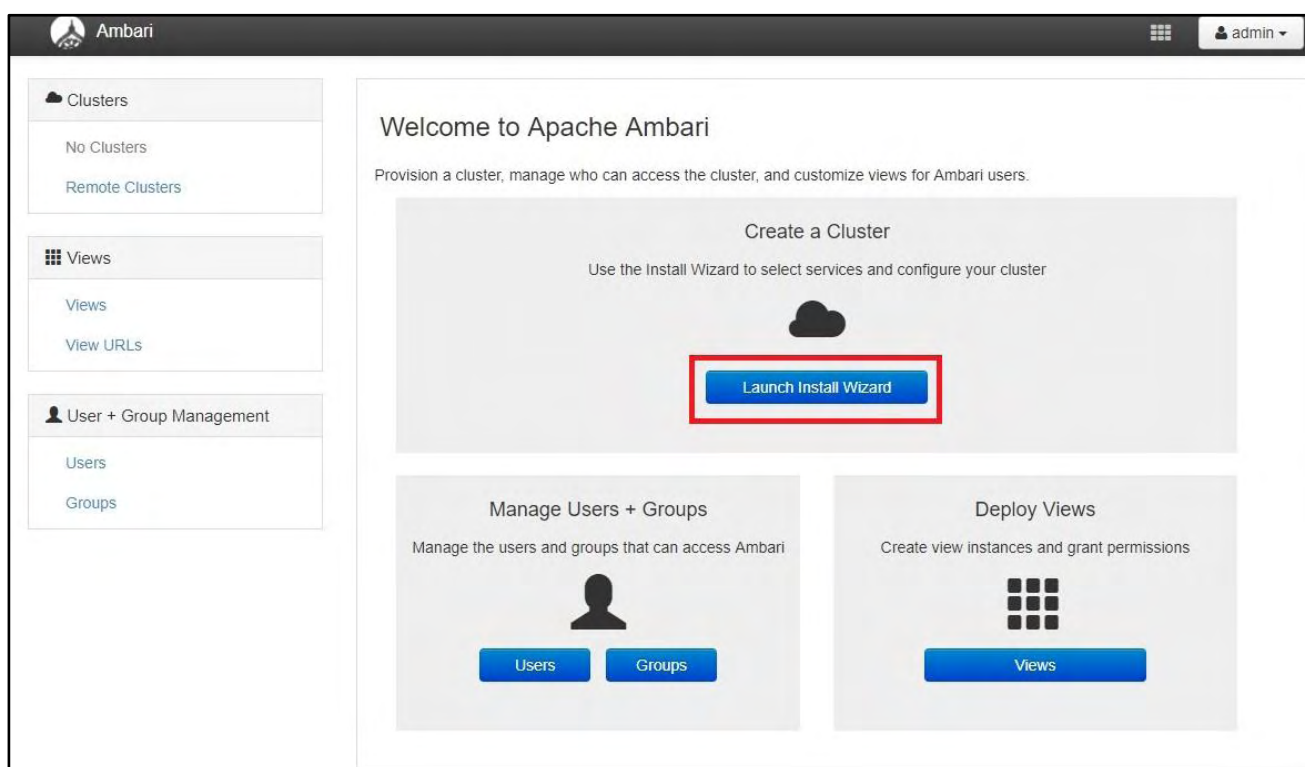
```
Found Ambari Server PID: 2627 at: /var/run/ambari-server/ambari-server.pid
```

8ο βήμα:

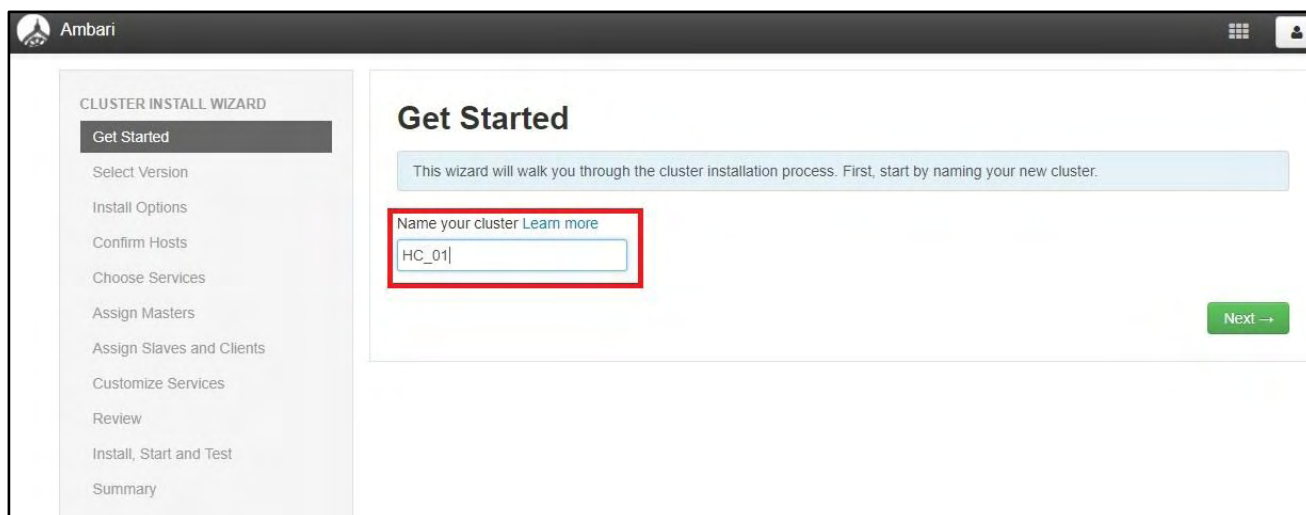
Η υπόλοιπη διαδικασία εγκατάστασης και ρύθμισης του cluster γίνεται μέσω web interface:



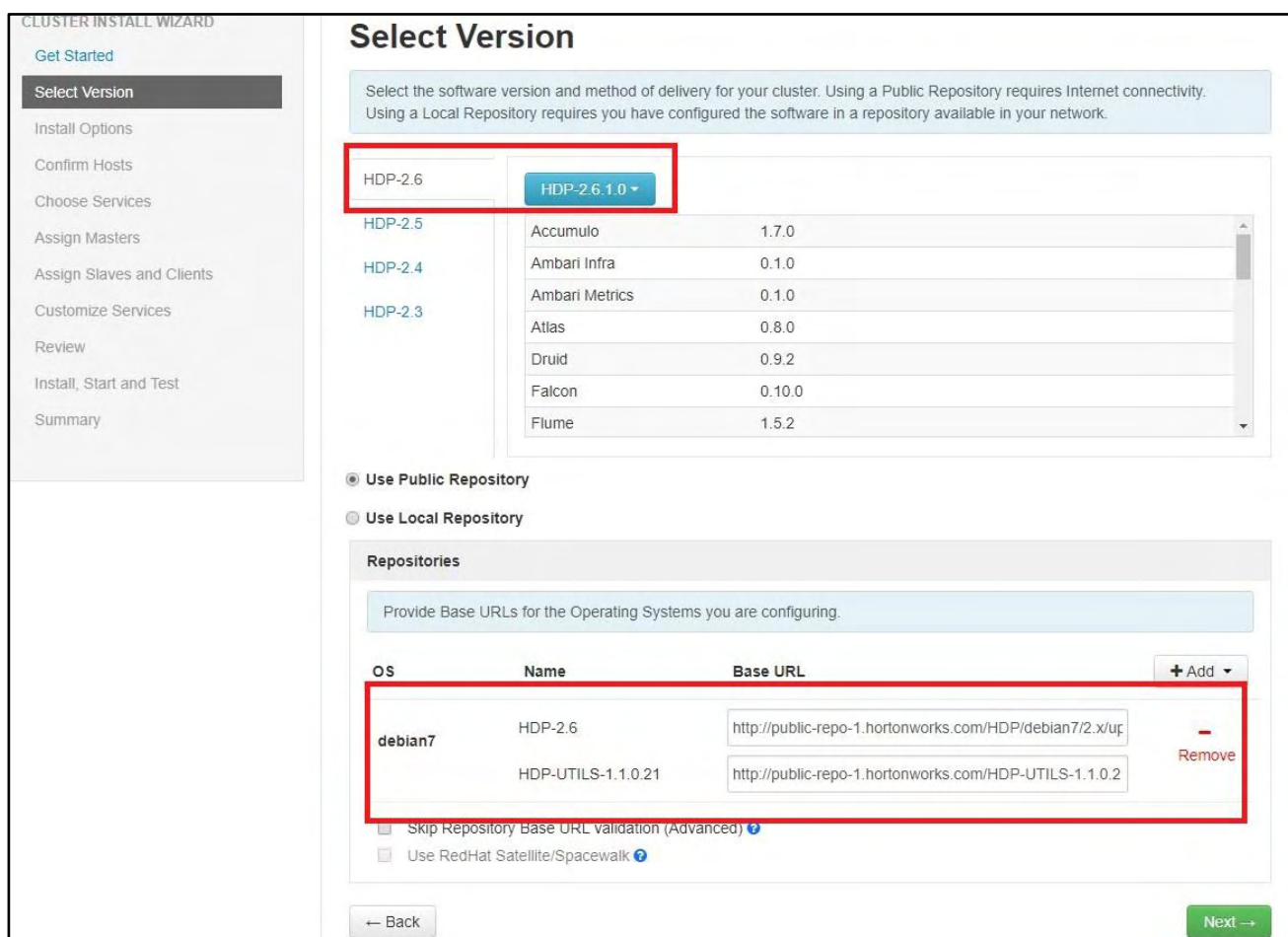
*Εικόνα 11: Σύνδεση στο Apache Ambari*



*Εικόνα 12: Εκκίνηση οδηγού εγκατάστασης*



Εικόνα 13: Επιλογή ονόματος cluster



Εικόνα 14: Επιλογή έκδοσης και αποθετηρίων



Εικόνα 15: Εισαγωγή ονομάτων servers και κλειδιού

Εικόνα 16: Επιβεβαίωση επικοινωνίας servers

Get Started

Select Version

Install Options

Confirm Hosts

**Choose Services**

Assign Masters

Assign Slaves and Clients

Customize Services

Review

Install, Start and Test

Summary

## Choose Services

Choose which services you want to install on your cluster.

| <input type="checkbox"/> Service                      | Version  | Description   |
|---|----------|---|
| <input checked="" type="checkbox"/> HDFS              | 2.7.3    | Apache Hadoop Distributed File System   |
| <input checked="" type="checkbox"/> YARN + MapReduce2 | 2.7.3    | Apache Hadoop NextGen MapReduce (YARN)  |
| <input type="checkbox"/> Tez                          | 0.7.0    | Tez is the next generation Hadoop Query Processing framework written on top of YARN.  |
| <input type="checkbox"/> Hive                         | 1.2.1000 | Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service  |
| <input type="checkbox"/> HBase                        | 1.1.2    | A Non-relational distributed database, plus Phoenix, a high performance SQL layer for low latency applications.   |
| <input type="checkbox"/> Pig                          | 0.16.0   | Scripting platform for analyzing large datasets   |
| <input type="checkbox"/> Sqoop                        | 1.4.6    | Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases   |
| <input type="checkbox"/> Oozie                        | 4.2.0    | System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the <a href="#">ExtJS</a> Library. |
| <input checked="" type="checkbox"/> ZooKeeper         | 3.4.6    | Centralized service which provides highly reliable distributed coordination   |
| <input type="checkbox"/> Falcon                       | 0.10.0   | Data management and processing platform   |
| <input type="checkbox"/> Storm                        | 1.1.0    | Apache Hadoop Stream processing framework   |
| <input type="checkbox"/> Flume                        | 1.5.2    | A distributed service for collecting, aggregating, and moving large amounts of streaming data into HDFS   |
| <input type="checkbox"/> Accumulo                     | 1.7.0    | Robust, scalable, high performance distributed key/value store.   |
| <input type="checkbox"/> Ambari Infra                 | 0.1.0    | Core shared service used by Ambari managed components.  |
| <input checked="" type="checkbox"/> Ambari Metrics    | 0.1.0    | A system for metrics collection that provides storage and retrieval capability for  |

Εικόνα 17: Επιλογή των βασικών υπηρεσιών Apache Hadoop

CLUSTER INSTALL WIZARD

[Get Started](#)
[Select Version](#)
[Install Options](#)
[Confirm Hosts](#)
[Choose Services](#)
[Assign Masters](#)
[Assign Slaves and Clients](#)
[Customize Services](#)
[Review](#)
[Install, Start and Test](#)
[Summary](#)

## Assign Masters

Assign master components to hosts you want to run them on.

SNameNode: 
NameNode: 
ResourceManager: 
App Timeline Server: 
History Server: 
ZooKeeper Server: 
ZooKeeper Server: 
ZooKeeper Server: 
Grafana: 
Metrics Collector: 
Activity Analyzer: 
HST Server: 
Activity Explorer:

hc-ambari.teiste.gr (9.8 GB, 6 cores)

NameNode
ZooKeeper Server
Grafana
Activity Analyzer
HST Server
Activity Explorer

hc-node01.teiste.gr (4.9 GB, 4 cores)

SNameNode
ResourceManager
App Timeline Server
History Server
ZooKeeper Server

hc-node02.teiste.gr (4.9 GB, 4 cores)

ZooKeeper Server
Metrics Collector

← Back

Next →

Εικόνα 18: Επιλογή των master servers

Ambari

CLUSTER INSTALL WIZARD

[Get Started](#)
[Select Version](#)
[Install Options](#)
[Confirm Hosts](#)
[Choose Services](#)
[Assign Masters](#)
[Assign Slaves and Clients](#)
[Customize Services](#)
[Review](#)
[Install, Start and Test](#)
[Summary](#)

## Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.  
 Hosts that are assigned master components are shown with \*.  
 "Client" will install HDFS Client, YARN Client, MapReduce2 Client and ZooKeeper Client.

| Host                  | all   none                                   | all   none                          | all   none                                      | all   none                                 |
|-----------------------|--|-------------------------------------|---|--|
| hc-ambari.teiste.gr * | <input checked="" type="checkbox"/> DataNode | <input type="checkbox"/> NFSGateway | <input checked="" type="checkbox"/> NodeManager | <input checked="" type="checkbox"/> Client |
| hc-node01.teiste.gr * | <input checked="" type="checkbox"/> DataNode | <input type="checkbox"/> NFSGateway | <input type="checkbox"/> NodeManager            | <input checked="" type="checkbox"/> Client |
| hc-node02.teiste.gr * | <input checked="" type="checkbox"/> DataNode | <input type="checkbox"/> NFSGateway | <input type="checkbox"/> NodeManager            | <input checked="" type="checkbox"/> Client |

Show: 25 1 - 3 of 3

← Back

Next →

Εικόνα 19: Επιλογή των slave servers

[Get Started](#)  
[Select Version](#)  
[Install Options](#)  
[Confirm Hosts](#)  
[Choose Services](#)  
[Assign Masters](#)  
[Assign Slaves and Clients](#)  
[Customize Services](#)  
[Review](#)  
[Install, Start and Test](#)  
[Summary](#)

## Customize Services

We have come up with recommended configurations for the services you selected. Customize them as you see fit.

HDFS
YARN
MapReduce2
ZooKeeper
Ambari Metrics 1
SmartSense 1
Misc

Group: Default (3)
Manage Config Groups
Filter...

Settings
Advanced

### NameNode

**NameNode**  
dfs.datanode.data.dir

/hadoop/...

**NameNode**  
dfs.datanode.data.dir

Determines where on the local filesystem an DFS data node should store its blocks. If this is a comma-delimited list of directories, then data will be stored in all named directories, typically on different devices. Directories that do not exist are ignored.

**NameNode**  
dfs.datanode.data.dir

1GB

0 GB 5 GB 9.782 GB

**NameNode**  
dfs.datanode.data.dir

150

1 101 200

**NameNode**  
dfs.datanode.data.dir

100%

99 % 99.5 % 100 %

### DataNode

**DataNode**  
dfs.datanode.data.dir

/hadoop/hdfs/data

**DataNode**  
dfs.datanode.data.dir

0

0 1

**DataNode**  
dfs.datanode.data.dir

1GB

0 GB 2.375 GB 4.88 GB

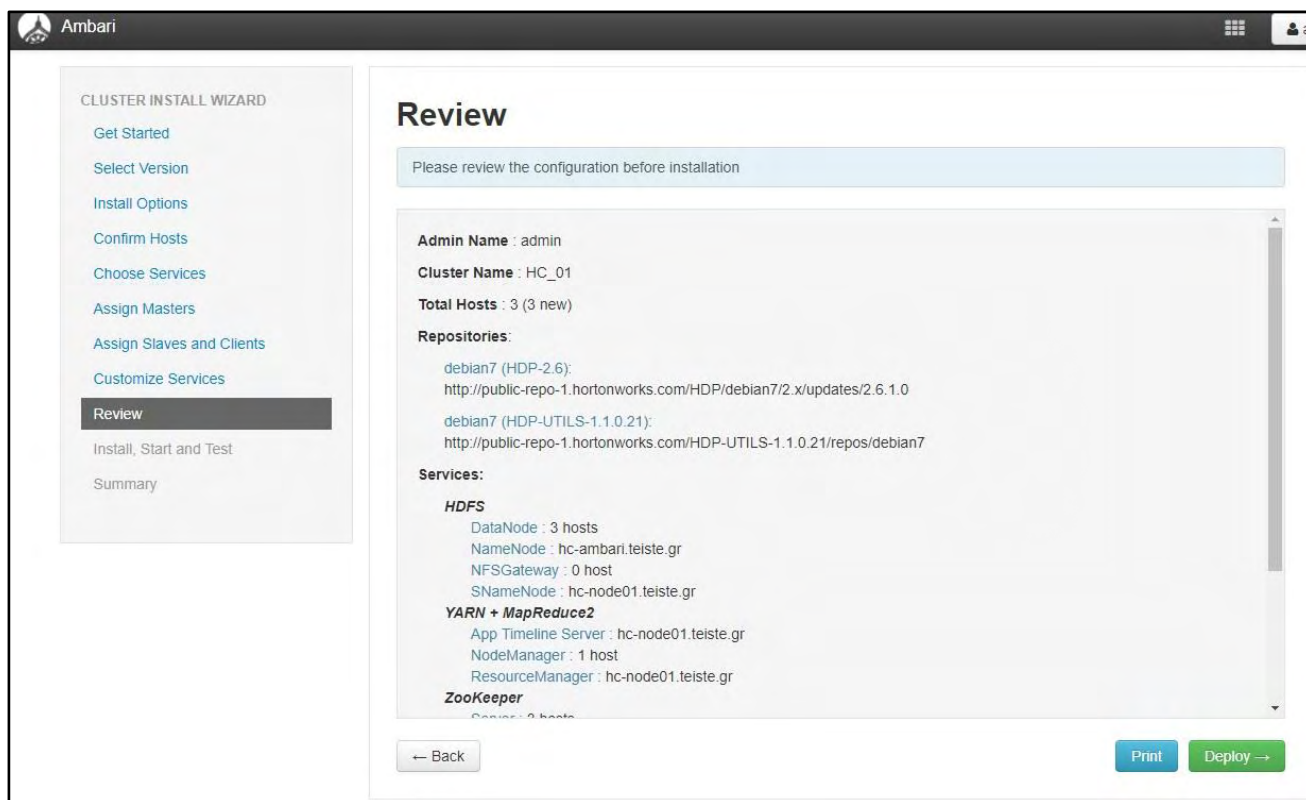
**DataNode**  
dfs.datanode.data.dir

4096

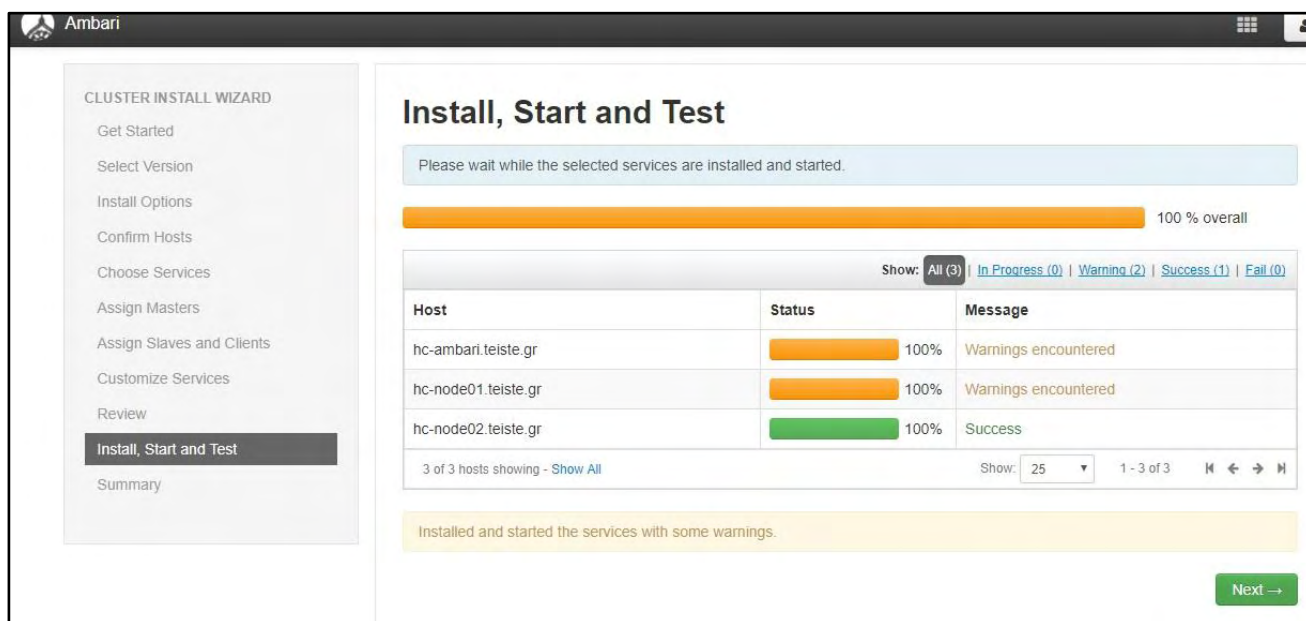
0 24000 48000

Εικόνα 20: Επιμέρους ρύθμιση παραμέτρων των υπηρεσιών





Εικόνα 21: Τελική οθόνη ρυθμίσεων



Εικόνα 22: Ολοκλήρωση εγκατάστασης

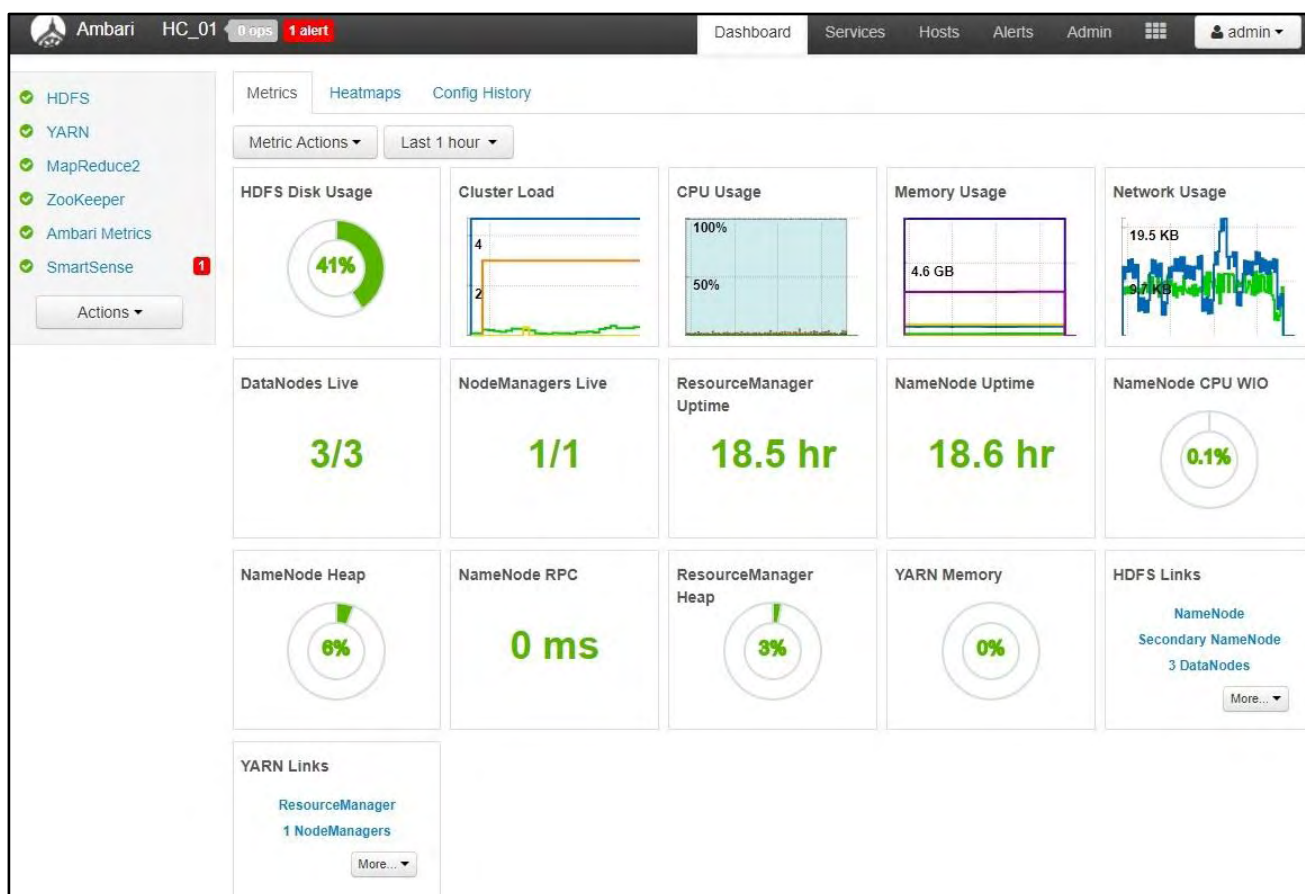
Με την παραπάνω διαδικασία ολοκληρώσαμε την εγκατάσταση του Apache Hadoop cluster με τη χρήση του εργαλείου Apache Ambari.

## Κεφάλαιο 3 :

### Διαχείριση Hadoop cluster με το εργαλείο Apache Ambari

Με το εργαλείο Apache Ambari μπορούμε να διαχειριστούμε το σύνολο των υπηρεσιών του cluster καθώς και τους servers που το αποτελούν. Επίσης, υπάρχει διαθέσιμο και ένα ολοκληρωμένο σύστημα ελέγχων και ειδοποιήσεων. Τέλος, μέσω γραφικού περιβάλλοντος (web interface) μπορούμε να διαχειριστούμε αρχεία και φακέλους του cluster. Παρακάτω παρατίθενται οι σχετικές εικόνες (screenshots) και τα αντίστοιχα σχόλια διαχείρισης του cluster.

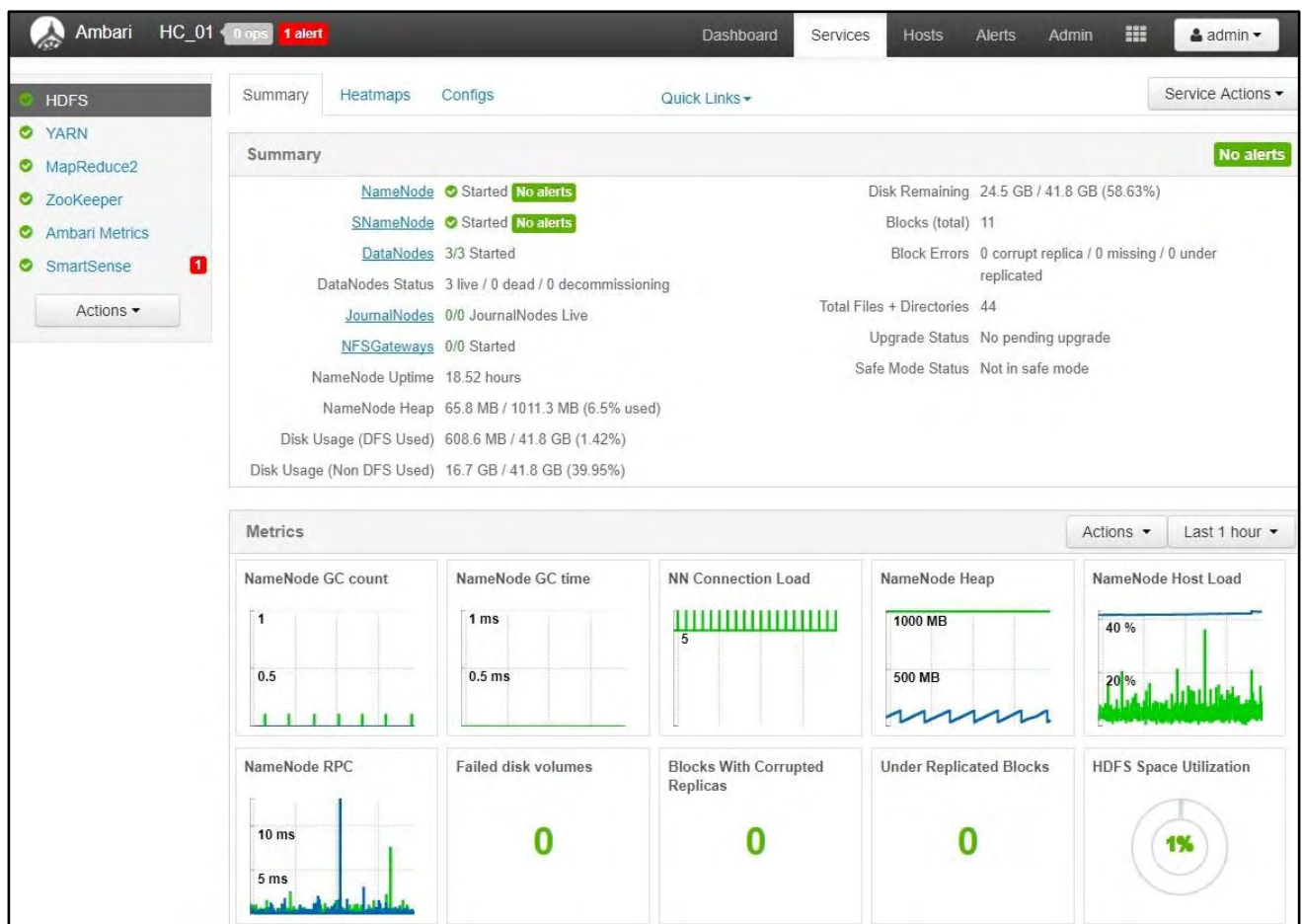
#### 3.1 Διαχείριση υπηρεσιών



Εικόνα 23: Κεντρικός πίνακας διαχείρισης υπηρεσιών

Στον κεντρικό πίνακα μπορούμε να δούμε τις τιμές των κυριότερων παραμέτρων cluster, όπως:

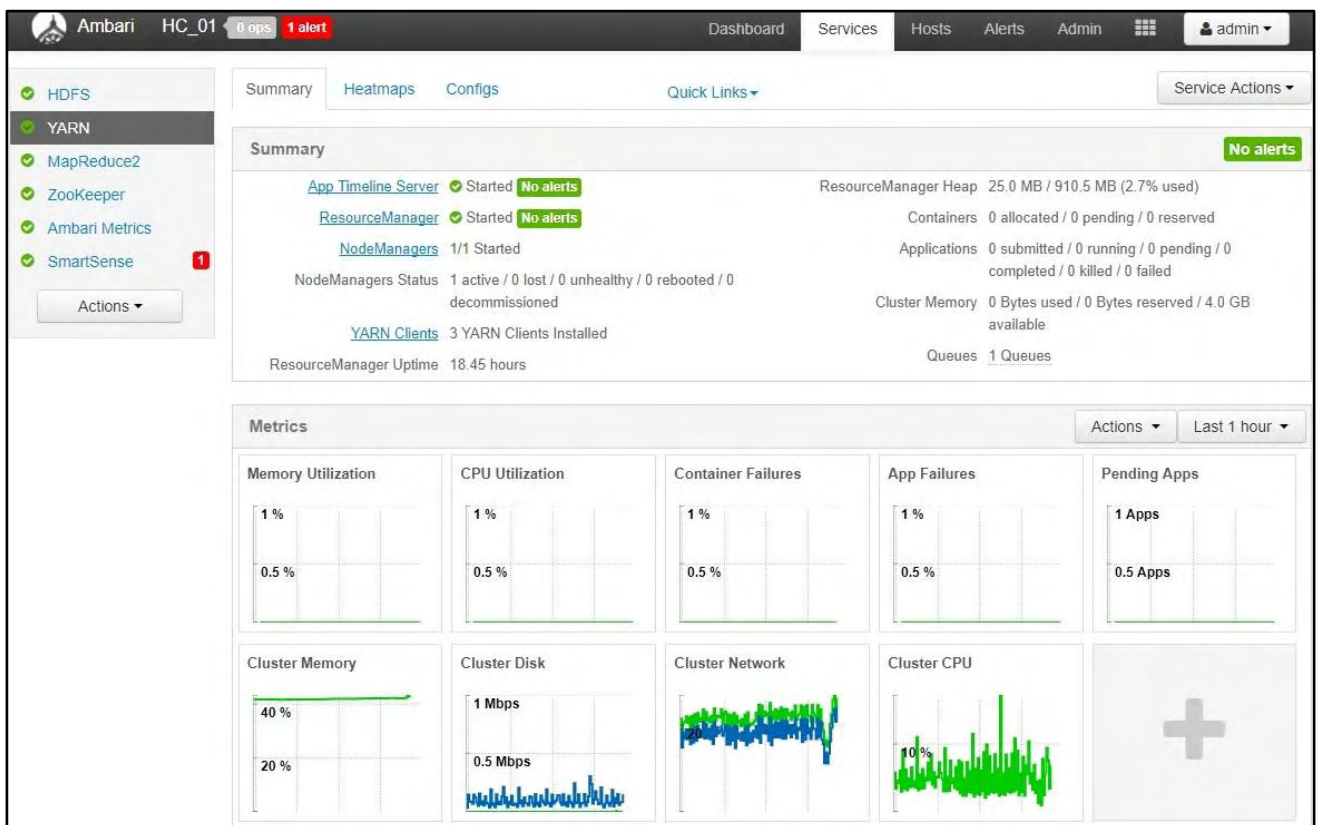
- τον ελεύθερο χώρο του συστήματος αρχείων (HDFS disk usage)
- το φόρτο του (cluster load)
- τη χρησιμοποιούμενη μνήμη (Memory Usage)
- τη χρησιμοποίηση του δικτύου (Network Usage)
- τη γενική κατάσταση των servers του cluster (DataNodes Live, NodeManager Live) καθώς και το χρόνο λειτουργίας τους (Uptime)



Εικόνα 24: Διαχείριση υπηρεσίας HDFS

Επιλέγοντας την καρτέλα HDFS, μπορούμε να δούμε με μεγαλύτερη λεπτομέρεια την κατάσταση της υπηρεσίας και των servers που είναι απαραίτητοι για τη λειτουργία της, όπως:

- ο NameNode (που έχει το βασικό ρόλο στο cluster),
- ο SNameNode (Secondary Name Node), που είναι εφεδρικός του NameNode,
- οι DataNodes servers (όπου αποθηκεύονται τα blocks των δεδομένων) και
- πληροφορίες σχετικές με την κατάσταση του συστήματος αρχείων

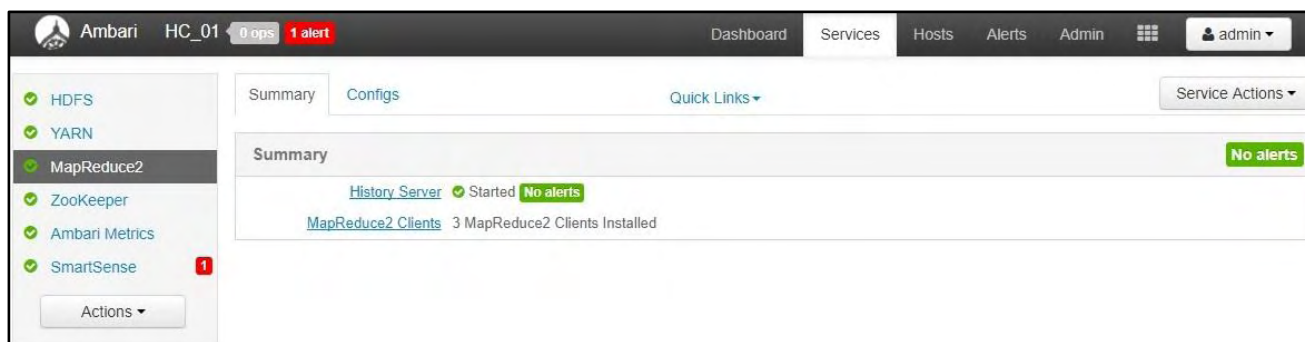


Εικόνα 25: Πίνακας διαχείρισης υπηρεσίας YARN

Επιλέγοντας την καρτέλα YARN, μπορούμε να δούμε με μεγαλύτερη λεπτομέρεια την κατάσταση της υπηρεσίας και των servers που είναι απαραίτητοι για τη λειτουργία της, όπως:

- ο App Timeline Server
- ο Resource Manager καθώς και
- πληροφορίες σχετικές με την κατάσταση των εργασιών του cluster (Applications, Cluster memory κτλ)





Επιλέγοντας την καρτέλα YARN, μπορούμε να δούμε με μεγαλύτερη λεπτομέρεια την κατάσταση της υπηρεσίας και των servers που είναι απαραίτητοι για τη λειτουργία της, όπως:

- ο History Server (όπου κρατείται ένα ιστορικό της λειτουργίας MapReduce) και
- η κατάσταση των servers που είναι εγκατεστημένη η υπηρεσία

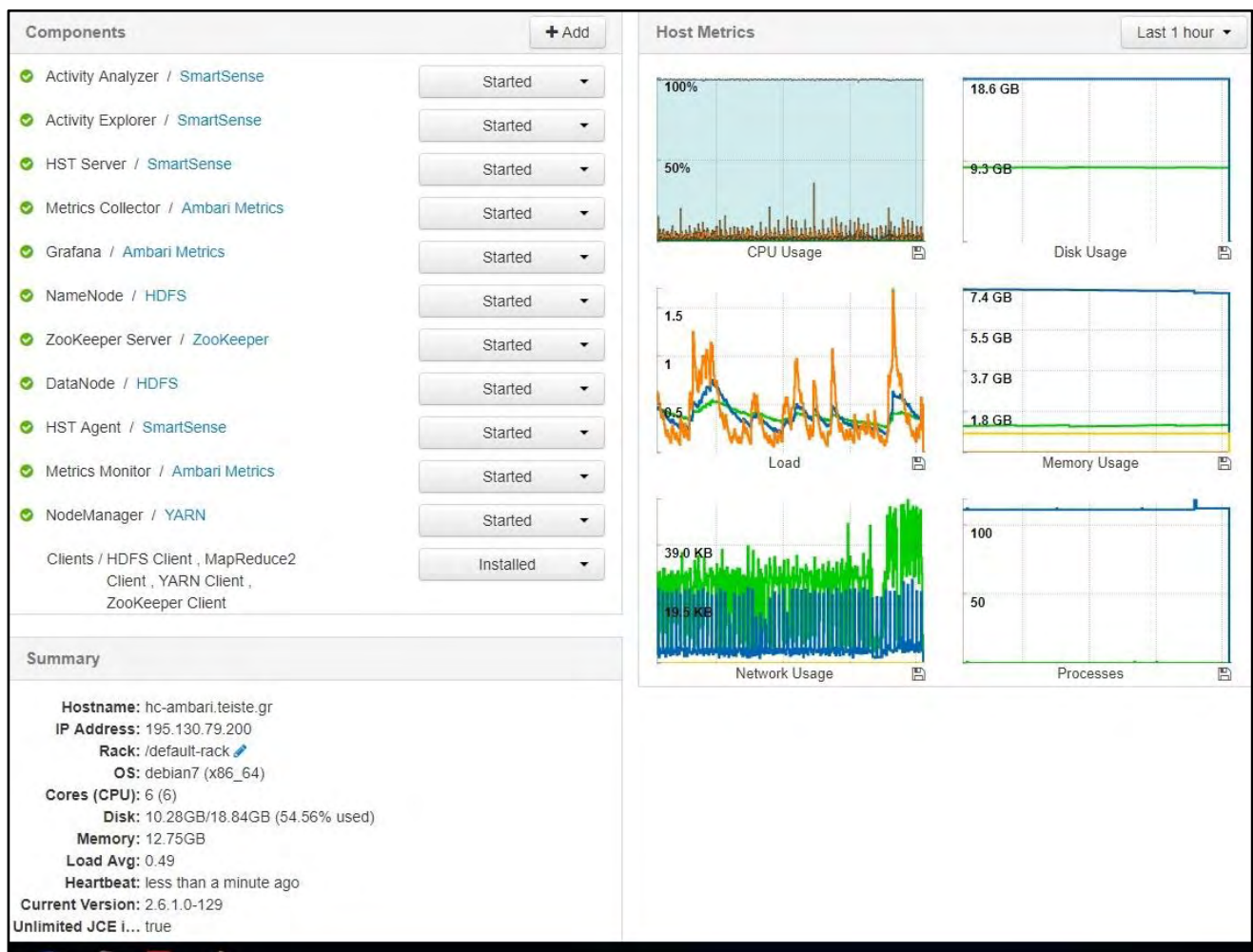
### 3.2 Διαχείριση σταθμών

| Name                | IP Address     | Rack          | Cores | RAM     | Disk Usage | Load Avg | Versions    | Components    |
|---------------------|----------------|---------------|-------|---------|------------|----------|-------------|---------------|
| hc-ambari.teiste.gr | 195.130.79.200 | /default-rack | 6 (6) | 12.75GB |            | 0.18     | HDP-2.6.1.0 | 15 Components |
| hc-node01.teiste.gr | 195.130.79.201 | /default-rack | 4 (4) | 4.86GB  |            | 0.02     | HDP-2.6.1.0 | 12 Components |
| hc-node02.teiste.gr | 195.130.79.202 | /default-rack | 4 (4) | 4.86GB  |            | 0.03     | HDP-2.6.1.0 | 8 Components  |

Εικόνα 26: Κεντρικός πίνακας διαχείρισης σταθμών

Επιλέγοντας την καρτέλα Hosts, μπορούμε να δούμε πληροφορίες των σταθμών (servers) που αποτελούν το cluster καθώς και γενικές πληροφορίες για τη χρησιμοποίηση των πόρων τους, όπως:

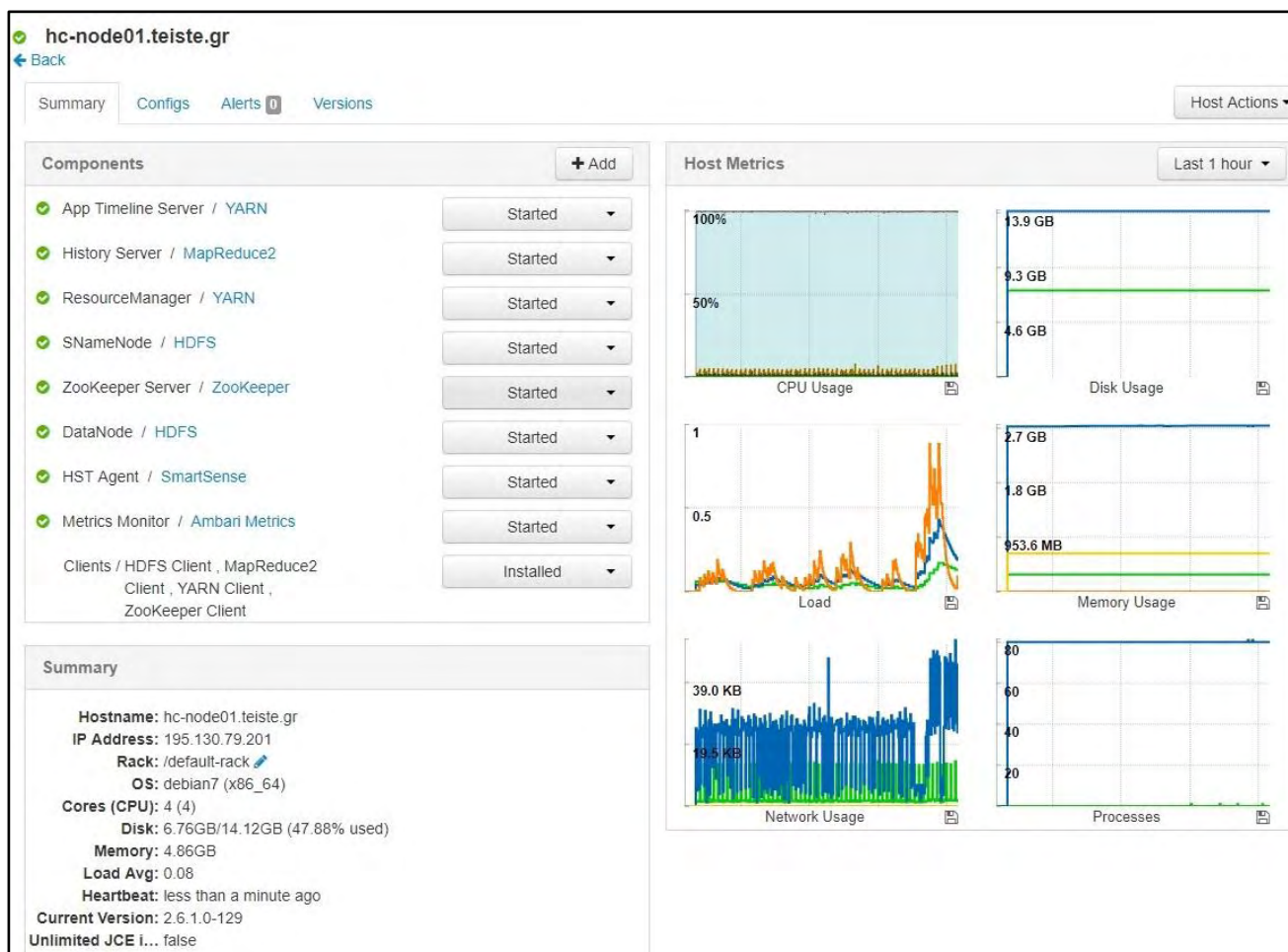
- Fully qualified domain name (FQDN),
- διεύθυνση IP,
- σε ποιο rack βρίσκονται (φυσική θέση),
- οι συνολικοί πυρήνες τους,
- η συνολική μνήμη RAM τους,
- ο φόρτος τους,
- την έκδοση Hadoop και
- τον αριθμό των υπηρεσιών που είναι εγκατεστημένες



Εικόνα 27: Πίνακας διαχείρισης Apache Ambari server

Επιλέγοντας το όνομα του server όπου είναι εγκατεστημένο το Apache Ambari (*hc-ambari.teiste.gr*), μπορούμε να δούμε με μεγαλύτερη λεπτομέρεια την κατάσταση του server καθώς την κατάσταση των υπηρεσιών που είναι εγκατεστημένες, όπως:

- στο πεδίο Components, η κατάσταση των υπηρεσιών,
- στο πεδίο Host Metrics, η κατάσταση των πόρων σε γραφικό περιβάλλον και
- στο πεδίο Summary, πληροφορίες του server



Εικόνα 28: Πίνακας διαχείρισης σταθμού cluster

Με παρόμοιο τρόπο μπορούμε να δούμε και πληροφορίες για τους υπόλοιπους σταθμούς του cluster.

### 3.3 Διαχείριση ελέγχου και ειδοποιήσεων

Μια από τις σπουδαιότερες λειτουργίες του Apache Ambari είναι το ολοκληρωμένο σύστημα ελέγχου (με ένα μεγάλο πλήθος από sensors) και ειδοποιήσεων που παρέχει στον διαχειριστή του cluster.

The screenshot shows the Ambari Alerts page for 'NameNode Host CPU Utilization'. The top navigation bar includes 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. The 'Alerts' tab is active, showing a list of alerts. The 'NameNode Host CPU Utilization' alert is selected, displaying its configuration and instances.

**Configuration**

**Description:** This host-level alert is triggered if CPU utilization of the NameNode exceeds certain warning and critical thresholds. It checks the NameNode JMX Servlet for the SystemCPULoad property. The threshold values are in percent.

**Check Interval:** 5 Minute

**Thresholds:**

| Level    | Value                 | Unit | Expression            |
|----------|-----------------------|------|-----------------------|
| OK       | {1} CPU, load {0..1%} |      |                       |
| WARNING  | 200                   | %    | {1} CPU, load {0..1%} |
| CRITICAL | 250                   | %    | {1} CPU, load {0..1%} |

**Connection Timeout:** CRITICAL 5 Seconds

**State:** Enabled

**Service:** HDFS

**Component:** NameNode

**Type:** METRIC

**Groups:** HDFS Default

**Last Changed:** Sat, Nov 25, 2017 18:47

**Check Count:** 1 (default)

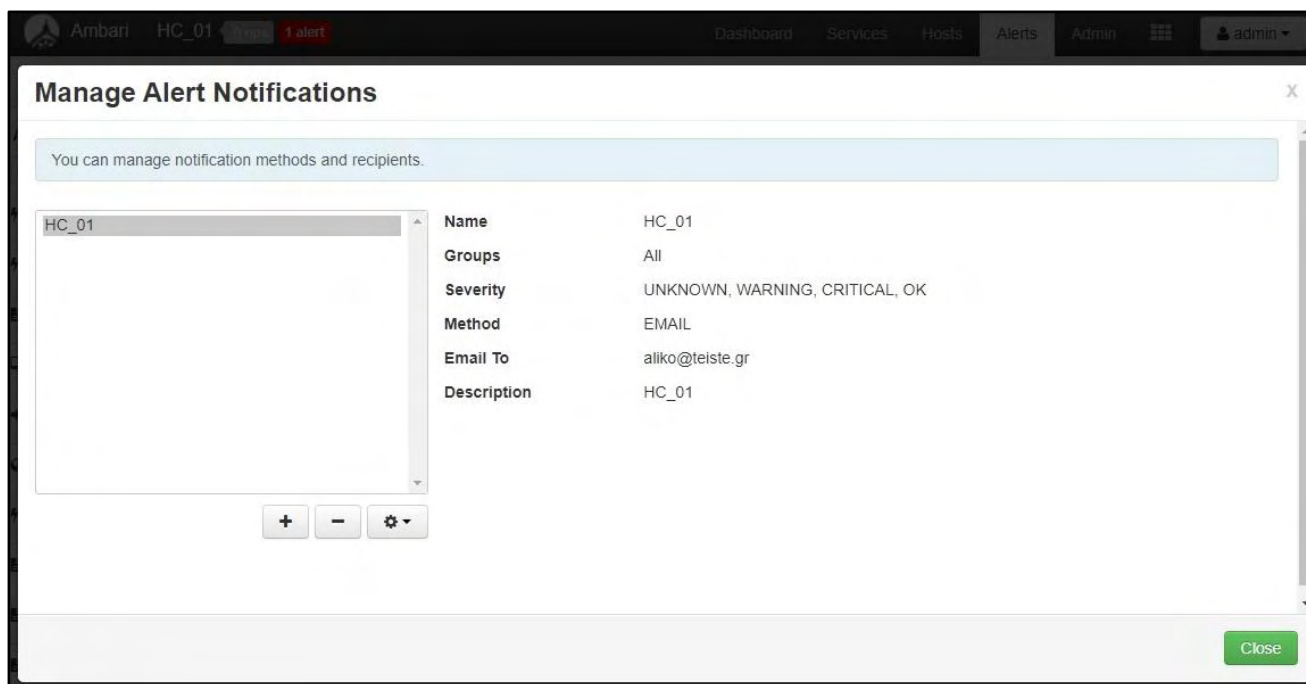
**Instances**

| Service | Host                | Status | 24-Hour      | Response |
|---------|---------------------|--------|--------------|----------|
| HDFS    | hc-ambari.teiste.gr | OK     | for 19 hours | 4        |

6 CPU, load 6.2%

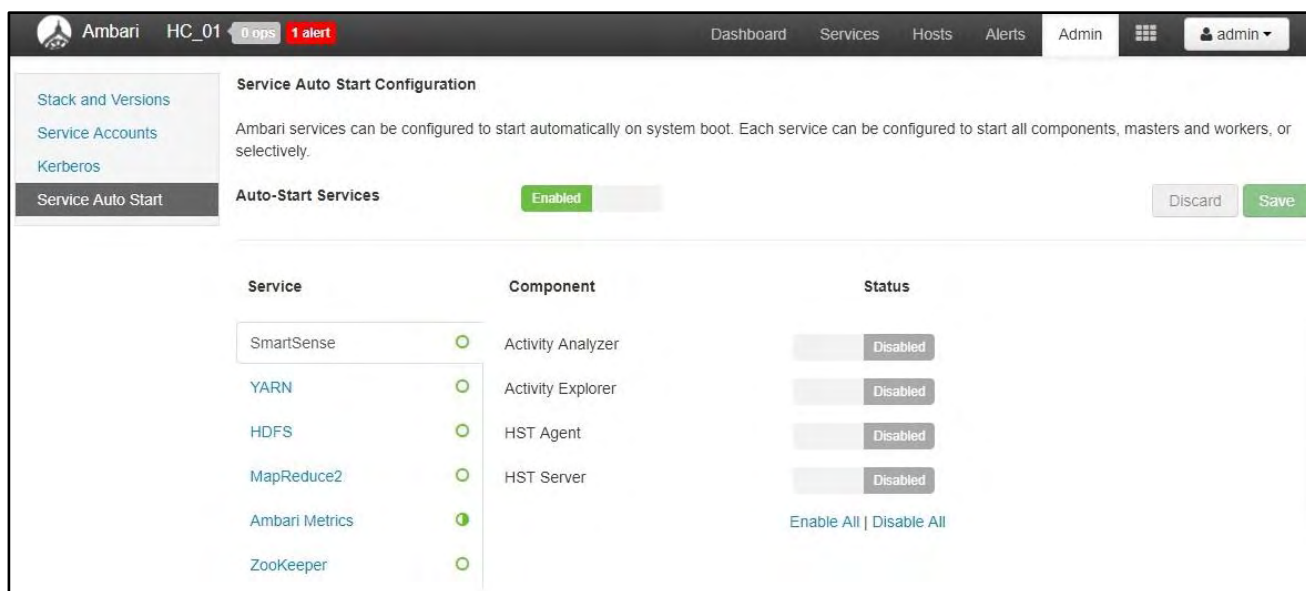
Εικόνα 29: Πίνακας διαχείρισης ειδοποιήσεων

Επιλέγοντας κάποιον sensor μπορούμε να δούμε τις αρχικές ρυθμίσεις του, οι οποίες σε περίπτωση που μας ικανοποιούν, μπορούμε με εύκολο τρόπο να τις τροποποιήσουμε κατάλληλα.



Εικόνα 30: Πίνακας διαχείρισης ειδοποιήσεων

Όπως αναφέραμε το Apache Ambari παρέχει ένα ολοκληρωμένο σύστημα ειδοποίησης μέσω e-mails σε έναν ή περισσότερους διαχειριστές που εύκολα μπορεί να τροποποιηθεί έτσι ώστε να υπάρχουν επίπεδα πρόσβασης στη διαχείρισή του cluster.

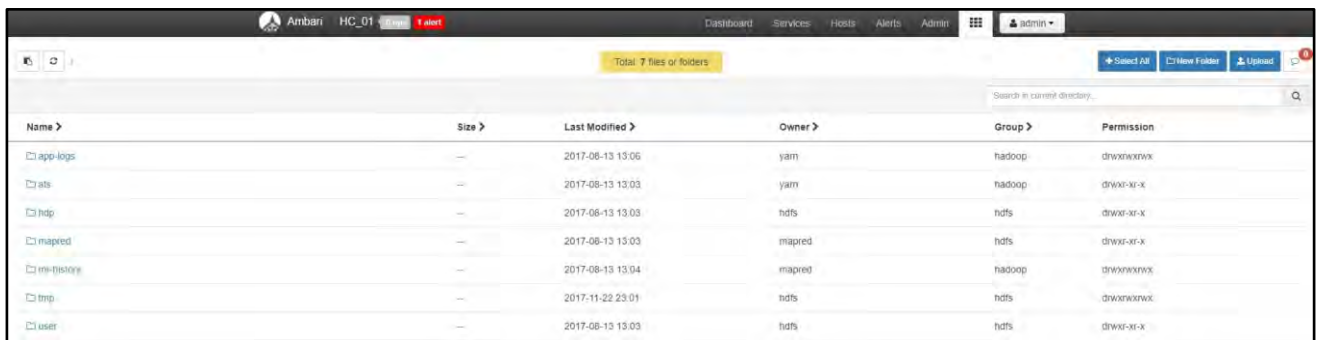


Εικόνα 31: Πίνακας διαχείρισης αυτόματης έναρξης υπηρεσιών



Επιλέγοντας την καρτέλα Admin/ Services Auto start, μπορούμε να επιλέξουμε ποιες υπηρεσίες θα ξεκινούν αυτόματα με την έναρξη του cluster (π.χ. μετά από επανεκκίνηση). Αυτή η δυνατότητα είναι πολύ σημαντική όταν στο cluster είναι εγκατεστημένες περισσότερες υπηρεσίες, που δεν είναι απαραίτητο να ξεκινούν αυτόματα αλλά να ενεργοποιούνται κατά επιλογή (on demand) του διαχειριστή.

### 3.4 Διαχείριση αρχείων και φακέλων



| Name >      | Size > | Last Modified >  | Owner > | Group > | Permission |
|-------------|--------|------------------|---------|---------|------------|
| app-logs    | —      | 2017-08-13 13:06 | yarn    | hadoop  | drwxr-xr-x |
| etc         | —      | 2017-08-13 13:03 | yarn    | hadoop  | drwxr-xr-x |
| hdp         | —      | 2017-08-13 13:03 | hdfs    | hdfs    | drwxr-xr-x |
| mapred      | —      | 2017-08-13 13:03 | mapred  | hdfs    | drwxr-xr-x |
| etc-history | —      | 2017-08-13 13:04 | mapred  | hadoop  | drwxr-xr-x |
| tmp         | —      | 2017-11-22 23:01 | hdfs    | hdfs    | drwxr-xr-x |
| user        | —      | 2017-08-13 13:03 | hdfs    | hdfs    | drwxr-xr-x |

Εικόνα 32: Πίνακας διαχείρισης αρχείων και φακέλων

Επιλέγοντας την καρτέλα με το σύμβολο του πλέγματος, μπορούμε με εύκολο τρόπο να διαχειριστούμε τους φακέλους (δημιουργία-διαγραφή) και τα αρχεία (ανέβασμα (upload)-κατέβασμα (download), μετονομασία) του cluster. Αυτή η δυνατότητα είναι πολύ σημαντική διότι επιτρέπει και μη εξοικειωμένους χρήστες να πραγματοποιούν βασικές λειτουργίες.

## Κεφάλαιο 4 : Επίλογος - Συμπεράσματα

### 4.1 Γιατί Hadoop

Υπάρχουν αρκετοί λόγοι για να επιλέξει κάποιος το Apache Hadoop για την επεξεργασία και την ανάλυση των δεδομένων. Παρακάτω είναι μερικοί από αυτούς:

- **Είναι οικονομικά αποδοτικό.** Το Apache Hadoop ελέγχει και μειώνει το κόστος αποθηκεύοντας τα δεδομένα πιο οικονομικά ανά Terabyte σε σχέση με άλλες πλατφόρμες. Αντί για χιλιάδες έως δεκάδες χιλιάδες δολάρια ανά Terabyte, το Hadoop παρέχει υπολογιστικές και αποθηκευτικές ικανότητες για εκατοντάδες δολάρια ανά Terabyte.

- **Είναι ανεκτικό σε σφάλματα.** Η ανοχή σε σφάλματα είναι ένα από τα πιο σημαντικά πλεονεκτήματα της χρήσης του Hadoop. Ακόμη και αν μεμονωμένοι κόμβοι έχουν υψηλά ποσοστά αποτυχίας κατά την εκτέλεση εργασιών σε ένα μεγάλο cluster, τα δεδομένα αναπαράγονται σε όλο το cluster, έτσι ώστε να μπορούν εύκολα να ανακτηθούν σε περίπτωση σφάλματος του δίσκου, του κόμβου ή του rack.

- **Είναι ευέλικτο.** Ο ευέλικτος τρόπος που τα δεδομένα είναι αποθηκευμένα στο Apache Hadoop είναι ένα από τα μεγαλύτερα πλεονεκτήματα του - επιτρέποντας στις επιχειρήσεις να δημιουργήσουν αξία και κέρδος από τα δεδομένα που μέχρι τώρα θεωρούνταν πάρα πολύ ακριβά για να αποθηκεύονται και να επεξεργάζονται σε παραδοσιακές βάσεις δεδομένων. Με το Hadoop, μπορούν να χρησιμοποιηθούν όλα τα είδη των δεδομένων, τόσο δομημένων όσο και αδόμητων, για την εξαγωγή πιο σημαντικών επιχειρηματικές ιδεών όσο πιο πολλά είναι τα δεδομένα.

- **Είναι επεκτάσιμο.** Το Hadoop είναι μια εξαιρετικά επεκτάσιμη πλατφόρμα αποθήκευσης, επειδή μπορεί να αποθηκεύσει και να διανείμει πολύ μεγάλα σύνολα δεδομένων σε cluster αποτελούμενα από εκατοντάδες φθηνούς server που λειτουργούν παράλληλα. Το πρόβλημα με τα παραδοσιακά συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), είναι ότι δεν μπορούν να κλιμακωθούν για την επεξεργασία τεράστιων όγκων δεδομένων.

## 4.2 Γιατί Apache Ambari

Με την αυξανόμενη δημοτικότητα του Hadoop, πολλοί χρήστες υιοθετούν την τεχνολογία αυτή για να διαπιστώσουν τις πολλές δυνατότητές του. Αλλά, επειδή γενικά η εφαρμογή του θεωρείται δύσκολη και πολύπλοκη, πολλοί χρήστες δεν μπορούσαν καν να περάσουν το φράγμα της εγκατάστασης του Hadoop. Πολλές διανομές προσφέρουν προεγκατεστημένο σύστημα Hadoop σε μορφή Virtual Machine για να δοκιμή, αλλά έτσι δεν δίνετε η αίσθηση του καταναεμημένου συστήματος. Επιπλέον, η εγκατάσταση ενός cluster δεν είναι εύκολη υπόθεση και με αυξανόμενο αριθμό κόμβων (servers) είναι πολύ δύσκολο να χειριστείς τόσες πολλές παραμέτρους διαμόρφωσης. Το εργαλείο Apache Ambari με το σύνολο των λειτουργιών του μέσα από ένα απλό γραφικό περιβάλλον (web interface) βοηθάει τους διαχειριστές να ξεπεράσουν όλα τα παραπάνω προβλήματα εγκατάστασης και διαχείρισης ενός Apache Hadoop cluster.

## 4.3 Επίλογος

Το Apache Hadoop, από τα πρώτα του βήματα μέχρι σήμερα, έχει εξελιχθεί σε όλους τους τομείς λόγω της ανάγκης εξαγωγής χρησιμοποιήσιμης πληροφορίας από τους τεράστιους όγκους δεδομένων που παράγονται κάθε μέρα και μέχρι τώρα έμεναν ανεκμετάλλευτοι.

Παρόλο τις αρχικές δυσκολίες υλοποίησης και διαχείρισής του Hadoop, πολλές εταιρείες έχουν δημιουργήσει αρκετές διανομές και εργαλεία όπως το Apache Ambari (που πραγματευόμαστε σε αυτή τη διπλωματική εργασία), που το έχουν κάνει πιο προσιτό και εύκολο στην εφαρμογή του.



## Πηγές - Βιβλιογραφία

- [1] What is Apache Hadoop?, (2012), <https://www.oreilly.com/ideas/what-is-apache-hadoop>
- [2] Hadoop, [http://www.sas.com/en\\_us/insights/big-data/hadoop.html](http://www.sas.com/en_us/insights/big-data/hadoop.html)
- [3] Welcome to Apache™ Hadoop®, (2014), <https://hadoop.apache.org/>
- [4] Hadoop: What it is, how it works, and what it can do, (2011), <https://www.oreilly.com/ideas/what-is-hadoop>
- [5] Apache Hadoop, <http://hortonworks.com/hadoop/>
- [6] Apache Hadoop, (2011), [https://en.wikipedia.org/wiki/Apache\\_Hadoop](https://en.wikipedia.org/wiki/Apache_Hadoop)
- [7] Apache Hadoop Basics, Hortonworks Inc., (2013)
- [8] Hadoop, The Definitive Guide, Tom White, (2015)
- [9] MapReduce, <https://en.wikipedia.org/wiki/MapReduce>
- [10] Apache Hadoop NextGen MapReduce (YARN), (2015), <https://hadoop.apache.org/docs/r2.7.1/hadoop-yarn/hadoop-yarn-site/YARN.html>
- [11] Hadoop in Action, Chuck Lam, (2011), ISBN: 9781935182191
- [12] <https://www.cloudera.com/>
- [13] <https://mapr.com/products/apache-hadoop/>
- [14] <https://hortonworks.com/>
- [15] <https://azure.microsoft.com/en-us/services/hdinsight/>
- [16] <https://wiki.apache.org/hadoop/PoweredBy#Y>
- [17] <https://wiki.apache.org/hadoop/PoweredBy#F>
- [18] <https://wiki.apache.org/hadoop/PoweredBy>
- [19] <https://www.linkedin.com/pulse/proxmox-virtual-environment-server-virtualization-kvm-tss-ravi-kumar>
- [20] <https://www.proxmox.com/en/>
- [21] <https://www.youtube.com/watch?v=cYEu0h61lsY>
- [22] <https://www.debian.org/releases/wheezy/amd64/>